Requirements for ASCI

JASON
MITRE

Requirements for ASCI

Study Leader
Roy Schwitters

Contributors Include:
Henry Abarbanel
Michael Brenner
John M. Cornwall
William Dally
Alvin Despain
Paul E. Dimotakis
Sidney Drell
Douglas M. Eardley
Robert Grober
Raymond Jeanloz
Jonathan Katz
Steven Koonin
Darrell Long
Dan Meiron (Consultant)
Francis Perkins
Christopher Stubbs
Peter Weinberger

October 2003

JSR-03-330

Approved for public release; distribution unlimited

JASON
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102-7508
(703) 883-6997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information estimated to average 1 hour per response, including the time for review instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget. Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE October 2003	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Requirements for ASCI

5. FUNDING NUMBERS

13039021-DC

6. AUTHOR(S)

Roy Schwitters, et al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The MITRE Corporation
JASON Program Office – W950
7515 Colshire Drive
McLean, Virginia 22102

8. PERFORMING ORGANIZATION REPORT NUMBER

JSR-03-330

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Department of Energy
National Nuclear Security Administration
1000 Independence Avenue, SW
Defense Programs, NA-114
Washington, DC 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

JSR-03-330

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

Distribution A

13. ABSTRACT *(Maximum 200 words)*

This is the report of the 2003 JASON summer study on the technical requirements for advanced scientific computing and modeling to support the United States maintaining a safe and reliable nuclear deterrent through the Department of Energy's (DOE) and National Nuclear Security Administration's (NNSA) science-based Stockpile Stewardship Program (SSP) Advanced scientific computing and modeling required by SSP is supported by the Advanced Simulation and Computing (ASCI) Program under SSP.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

Contents

1 EXECUTIVE SUMMARY

This section summarizes the conclusions and recommendations of the 2003 JASON summer study commissioned by the National Nuclear Security Administration (NNSA) to identify the distinct requirements of its stockpile stewardship program (SSP) in relation to the hardware procurement strategy of the Advanced Simulation and Computing (ASCI) program. In particular, we were tasked to evaluate the increased risk to the nuclear weapons (NW) stockpile and the scientific program of SSP as a result of delaying computer acquisitions intended to advance computing capability. We were also charged to consider the confidence in our NW simulation capability and the appropriate balance between near-, intermediate- and long-term SSP needs for acquiring new hardware with increased computing capability.

Today, ASCI comprises high-performance computing hardware, suites of large codes built on a validated scientific/engineering base, experienced people and connections to the greater scientific, computing and national security communities. The tools and methods developed under ASCI have evolved to the point where they are today *essential* to stockpile stewardship. So are the people supported through ASCI who, working closely with NW designers, engineers and managers, have acquired invaluable expertise in developing and optimizing ASCI tools and in establishing and improving the scientific credibility of NW simulations. Some notable ASCI accomplishments are described in the body of this report.

Two commonly used measures of the overall productivity of ASCI platforms are *Capability* and *Capacity*. The common unit of measure for both is peak floating-point operations, noted as TeraFlops or TF (10^{12} operations per second). Used in this context, *Capability* refers to the maximum processing power possible that can be applied to a single job and *Capacity* represents the total processing power available from all machines capable of operating ASCI codes. A given amount of *Capability* implies *Capacity* in two ways: 1) by its direct contribution to total capacity and 2) because a high-*Capability* machine can be re-configured into multiple lower-*Capability* machines to run multiple shorter jobs, often with somewhat improved overall performance.

Today, the ASCI platforms of highest *Capability* are LLNL's "White" at 12.3 TF and LANL's "Q" at 20 TF. The next planned acquisitions are SNL's "Red Storm" projected to be 40 TF and LLNL's "Purple C" at 100 TF. SSP requirements over the next few years call for a few large jobs which need the largest available *Capability* but the most important trend is a factor-of-two oversubscription in ASCI *Capacity* (well supported in our view by distinct technical requirements) which is projected to go on and potentially worsen in the foreseeable future. This level of demand means that jobs are selected to run by some combination of administrative fiat and management priority-setting; it creates strong incentives for all involved in ASCI computing to improve performance of algorithms and platforms for greater delivered performance. A strong and valuable effort has been made by the ASCI program to increase the efficiency of performance (ratio of computing operations delivered to peak performance) to current levels; in practice, about 0.5–15% of the peak processing speed is realized, depending on algorithms and details of implementation. Similar efficiencies are found in many commercial, engineering and scientific applications.

Within 10 years, estimates of the demand for *Capability* and general physics arguments indicate a machine of 1000 TF = 1 PetaFlop (PF) will be needed to execute the most demanding jobs. Such demand is inevitable; it should not be viewed, however, as some plateau in required *Capability*—there are sound technical reasons to expect even greater *Capability* demand in the future.

We were charged to evaluate the increased risk to the stockpile and to the scientific program it supports that would result from delaying acquisitions to advance *Capability*. To assess this additional risk, we constructed an acquisition scenario where FY 04 funding was reduced by $33M (24%), requiring that the procurements of Red Storm and Purple C be stretched out. Given the approximately $50M shortfall in requested FY 03 funds, the FY 04 reduction was assumed to lead to further reductions in FY 05, 06 and 07, which we modeled by an approximately flat acquisition budget through FY 08 at a level of approximately $120M–$130M. The net effect of such a stretch-out is to reduce overall *Capacity* to below 1/3 of demand during the critical program years FY 05–08. In this period, several program milestones are to be completed including the refurbishment of one major weapons system, and the qualification of critical components of another. We judge the risk to SSP of such a delay to be *high*, not so much from the delayed *Capability*, but from the very serious reduction in overall *Capacity*. The large jobs projected to require 100 TF-level *Capability* might be deferred at moderate risk to the program, but the resulting very large oversubscription in *Capacity* risks becoming unmanageable. Purchasing the proposed large platforms—Purple C and Red Storm—on a stretched-out, suboptimal schedule where their CPUs and other components are bought after their performance/cost prime strikes us as being unwise both in terms of delivered capability and capacity.

In evaluating NNSA's current ASCI platform acquisition strategy, we see two areas of substantial risk. The first is related to the current and projected oversubscription in *Capacity*. A factor-of-two oversubscription is probably manageable; however, larger demand on *Capacity* could become unmanageable. The likely result would be the delay in meeting SSP technical milestones and/or overly-cautious decision-making leading to expensive—at the $50M–$100M level—mitigation programs from elsewhere in the SSP (for addressing SSP life-extension-program or significant-finding-investigation issues, for example) that might not be necessary if sufficient confidence were provided by timely simulations. The second area of risk is the lack of a credible "road map" to acquiring the next generation of machines with PF *Capability*, which will be needed within a decade. Scaling to 1 PF using present machine architectures implies very large numbers of processors—of order 100,000, perhaps—might be needed. Such large numbers raise serious questions of scalability of code performance and of machine reliability.

JASON recommends SSP management consider four general areas for mitigating the risks associated with its present ASCI acquisition strategy:

1. Platform Acquisition:

   (a) Modify the allocation of resources in the current acquisition plan to provide additional *Capacity* platforms starting as soon as possible.

   (b) Lay the groundwork for future *Capability* machines. This may involve acquisition of "*Capability*-exploration" machines focused on optimizing efficiency in computation for ASCI problems in order to gain experience with architectures that might plausibly be extended to the PF level. These acquisitions can probably begin in the FY 06–07 time frame and may be able to replace currently planned greater-than 100 TF-scale machines that are scheduled to follow the Purple C and Red Storm acquisitions.

2. SSP Requirements:

   Set priorities in the SSP requirements and assign ASCI resources accordingly. This is essential to reduce excess computing demand and to assure that the high priority problems are addressed to meet goals as scheduled. In particular, some of the stockpile-to-target-sequence requirements identified for Cold War scenarios place significant demands on ASCI resources (and, in fact, on the entire SSP). They should be reviewed carefully in the context of current and anticipated US security needs. Those judged to no longer meet a compelling cost/benefit standard should be either relaxed or assigned an appropriately lower ranking in the queue of high-priority tasks for ASCI's available and planned future resources.

3. ASCI Operations:

   (a) Expand access to ASCI "most-capable" systems to best align ASCI *Capability* with overall SSP priorities and needs. We applaud efforts to make ASCI platforms available to workers across the complex, regardless of where a given machine or work-group is located, but present quotas should be re-examined. We expect additional benefits to accrue from enhanced scientific communications and competition. Depending on the architecture of future PF-level machines, it is possible that future acquisitions will entail fewer high-*Capability* platforms, in which case convenient and flexible access across the complex becomes essential.

(b) Continue and expand, as appropriate, investment in computational science investigations directed toward improving the delivered performance of algorithms relevant to ASCI; consider dedicating regularly scheduled machine time on capability platforms for efficiency studies.

4. Nuclear Weapons Science:

   Encourage the advance of NW science at every opportunity in the SSP and ASCI programs. Better science is the most cost-effective way to reduce risk in the stewardship program and the only possible way to achieve sufficiency in the modeling and understanding of these complex systems. Some excellent new science is beginning to emerge in association with ASCI. In the body of this report, we comment on some of this new science and suggest possible extensions.

To summarize, our major concern is in providing as soon as practical much needed capacity to the ASCI program lest the resulting very large oversubscription in *Capacity* becomes unmanageable. In addition, a road map must be developed to deliver to the program machines of the requisite *Capability*.

2 INTRODUCTION

This is the report of the 2003 JASON summer study on the technical requirements for advanced scientific computing and modeling to support the United States maintaining a safe and reliable nuclear deterrent through the Department of Energy's (DOE) and National Nuclear Security Administration's (NNSA) science-based Stockpile Stewardship Program (SSP). Advanced scientific computing and modeling required by SSP is supported by the Advanced Simulation and Computing (ASCI) Program under SSP.

The Senate Energy and Water Development Appropriations Committee included the following language in the Fiscal Year 2003 appropriations bill: "The NNSA is directed to commission two related studies, the first to be performed in collaboration with the Department's Office of Science and the second focused solely on issues relevant to the stockpile stewardship program. These studies should address issues of alternative computer architectures and the requirements that drive them."

NNSA charged JASON to conduct the second of the studies directed in the committee language. The formal charge made to JASON by NNSA is given in the following section.

2.1 Charge

Identify the distinct requirements of the stockpile stewardship program and its relation to the ASCI acquisition strategy.

This study should describe the requirements of the stockpile mission and the acquisition of computing capability and capacity. The study must consider confidence in our simulation capability and the appropriate balance between capacity drivers, the near-, intermediate- and long-term DSW demands. As time passes, the devices in the enduring stockpile continue to age and increasingly complex challenges emerge to confront our simulation capability. Therefore, for a sustainable and credible program over the long term, attention must be paid to the scientific basis, verification and validation of the codes and a tractable transition to three-dimensional simulations.

Evaluate the increased risk to the stockpile and to the scientific program which it supports, as a result of delaying acquisitions to advance capability. Meeting the requirements identified in the first part of the study is the essential responsibility of the program. The near-term requirements associated with the annual assessment and certification plus addressing significant finding investigations (SFIs) that surface as a result of regular surveillance must be balanced against the long-term goal of reduced phenomenology and increasingly credible predictive capability.

Finally, this report shall include an evaluation of the trade-offs between strategies that accelerate or postpone acquisition of capability, as contrasted from capacity computing.

2.2 Summer Study

One-day site visits were arranged during Spring 2003 to the three weapons laboratories, Los Alamos National Laboratory (LANL), Lawrence Livermore National Laboratory (LLNL) and Sandia National Laboratory (SNL) to give us a sense of the actual working environment in the ASCI program. At each lab, several JASONs were able to see ASCI facilities and interact informally with designers, programmers and managers who use the ASCI systems. This was a most valuable experience. We are especially grateful for the time and effort put in by many people at the labs that made these visits so helpful.

The summer study began in June with 5 1/2 days of briefings by laboratory experts and NNSA staff on all aspects of the ASCI program, emphasizing scientific results, technical requirements for computing and working experience. Particulary helpful to us during these briefings was a panel discussion by young designers from the labs who were probed for their practical experience and frank opinions working with ASCI systems.

Following the technical briefings, we held a half-day informal discussion with the three laboratory weapons system associate directors to hear their perspectives on the ASCI program. We also met with several nationally recognized experts in high-performance computing, outside ASCI, for their input. A full list of the briefers who contributed to this study is given in the appendix.

Follow-up questions were presented to the labs following the briefings; their responses to these questions, published material on ASCI and the full body of briefings and discussions conducted during the study formed the basis for this report.

2.3 The ASCI Program

The mission of the ASCI Program is to provide the leading edge, high-end simulation capabilities needed to meet weapons assessment and certification requirements of the SSP. The specific objectives are: To create science-based, predictive simulations of nuclear weapon systems to analyze behavior and to assess weapon performance, in an environment without nuclear testing. To predict with high certainty the behavior of full weapon systems in complex accident scenarios, to help ensure safety. To achieve sufficient, validated predictive simulations to extend the lifetime of the stockpile, predict failure mechanisms, and reduce routine maintenance. To use virtual prototyping and modeling to understand how new production processes and materials affect performance, safety, reliability, and aging.

These simulations must be based upon a sufficiently good scientific understanding of material properties and physical processes, in the extreme ranges of temperature, pressure and timescale relevant to nuclear weapons. This understanding comes from application of the known laws of physics, from experiments, and from the archive of nuclear test data.

The ASCI "codes" typically model a weapon by numerically dividing it into a very large number (up to billions) of interacting gridpoints. Basic processes are simulated at each gridpoint in discrete timesteps, and nearby gridpoints interact by exchanging mass or energy. Hydrodynamics, material motion, heat diffusion, nuclear reactions, and other processes are simulated in this way. Such a grid maps reasonably well into a massively parallel computer, or a vector computer, but due to the intricacy of the algorithms, the fit is far from perfect, and realistic performance is generally much less than "theoretical peak FLOPS". For transport of radiant energy and of neutrons, each gridpoint may also need to track many groups in direction and in particle energy, further multiplying the computing requirements; here the fit onto existing computer architectures is even less perfect.

Acquisition of large-scale computers amounts to about 15% of the total ASCI Program budget, or about $94M/yr (FY00–FY02 average; the figure for FY03 is $51M). As technology improves, new large machines are bought. Currently the most capable is the "Q" machine at LANL, with a theoretical peak performance of 20 TFLOPS (TF) (trillions of floating point operations per second). Future plans call for a 40T FLOPs machine (SNL Red Storm in 04) and a 100T FLOP (LLNL Purple C in 05). In addition to these computers with the greatest capability, other smaller ones are needed to supply bulk capacity computing to users.

ASCI computers are heavily used, indeed oversubscribed, by the designers, physicists and engineers, and are absolutely essential to the SSP. Yet current ASCI computers and codes, while extraordinarily powerful, do not yet fully meet the program objectives, because better resolution is needed (more gridpoints and timesteps) to resolve important processes which take place on tiny length scales, and also because our physical understanding of some processes, notably turbulence and mix, is still not good enough. (The latter processes will need sub-grid modelling.) Therefore continued acquistion of suitable, more powerful, computers is planned, following the advance of technology.

3 ASCI PROGRAM REQUIREMENTS

3.1 Components of the Stockpile Stewardship Program and Their Demands on ASCI

The ASCI program is driven by a number of science-based stockpile stewardship (SBSS) requirements and milestones. These include:

1. Directed Stockpile Work (DSW). DSW involves evaluation, maintenance, and refurbishment of the stockpile. The ultimate output of DSW comes through the annual certification process, which specifies that each weapon in the stockpile is usable or not. The computing power needed to meet various DSW priorities varies over a large range. A major component of DSW is the set of Life Extension Programs (LEPs), which deal with the discovery and/or prediction of aging problems in specific weapons, and the problems of refurbishing these weapons in view of the aging and other problems, through the design, testing, and installation of new components.

2. Campaigns. These organize the science-based stewardship effort into various functional areas (primaries, secondaries) and science or engineering areas. Campaigns involve theory and experiment, simulations, and surveillance. Campaigns call for ASCI-related milestones. These are extremely demanding on computer resources, and explore the limits of ASCI capability.

3. Significant Finding Investigations (SFIs). Defects in specific stockpile weapons may be found as a result of the standard surveillance process, in which a number of weapons are disassembled and inspected each year. If the defect is considered serious enough to affect the reliability of the weapon type, an SFI is opened. An SFI may involve leaks, loose parts or wires, and similar defects; such defects, to the extent that they are not simply random individual defects, may trigger the SFI. Ordinarily, although computing may be very helpful in resolving any particular SFI, the computing demands are not among the biggest. However, this depends greatly on the nature of the SFI.

4. Baselining. Baselining is one of the most computer-intensive activities in the whole stewardship program. It is an attempt to find a unified and integrated view of a great many underground tests (UGTs) through simulations, with the goal of reducing (ultimately to zero) special phenomenologies which may

differ from UGT to UGT. The computing load here may be enormous, since potentially hundreds of tests might be simulated, with many runs needed to verify phenomenology for any given test.

5. Safety. ASCI supports safety requirements through simulations of drop tests, fires and other abnormal environments. While present-day safety calculations go far beyond what was possible a decade ago, they need not be the most challenging simulations called for in the ASCI program. In many cases experiments are quite feasible and preferable to calculations.

6. STS requirements. The Stockpile-to-Target Sequence (STS) requirements specify what environments a given weapon type will encounter throughout its life, culminating in actual use. These include temperature excursions and hostile environments, such as might be encountered through enemy nuclear defense missiles or fratricide. In many cases, the STS requirements stem from the Cold War, and may not represent a realistic assessment of the kind of environment any particular weapon system may encounter. Some STS requirements are extremely stressing on quantifying the margins and uncertainties of the weapon, and end up driving some heroic ASCI simulations.

7. Support to the production complex. ASCI is used for simulations related to assembly and disassembly at the plants, and also for design of better tools and processes. The computing loads are not usually unduly demanding.

8. Surety. This includes a variety of tasks, some of which we cannot discuss here. Some of these tasks involve use control of US weapons. It is claimed that certain tasks require ASCI capability resources, but we have not seen the evidence to support this claim.

9. Advanced concepts. Advanced concepts need to be studied for a number of reasons, including keeping designers intellectually sharp and up to date, to understand possible foreign advances in nuclear weapons design, as well as to respond to perceived new needs for the stockpile. It is certainly possible that the demands on ASCI resources would be high, but this does not appear to be the case now.

10. Weapons science. The science here includes materials properties such as strength at pressures not attainable in the laboratory; behavior of high explosives; physics of radiation transport; and a host of others. In many cases, understanding of basic phenomena, whether through theory or experiment, is limited. ASCI demands are often high, because of the complexity of the phenomena being modeled.

One sometimes hears it said that SBSS is too much a collection of nice-to-know science projects and not oriented enough to program requirements which refer to specific weapons and their problems. Would it not, the argument goes, reduce the strain on ASCI resources and acquisitions to emphasize the computing load for DSW, SFIs, and the like, and de-emphasize the computing load assigned to better understanding of weapons science? The reasoning would be that among the science-based computing loads is baselining, which demands by far the greatest share of ASCI resources.

There are two answers to this. The first is that baselining is considered by the weapons laboratories to be essential to such DSW as the LEPs. They argue that one can hardly have confidence in a program of refurbishment and evaluation of aging problems without confidence in the underlying weapons science which is to be buttressed by successful baselining of the UGTs associated with the LEP in question. There is much truth in this position. However, one should keep in mind that there may be, in some cases, more direct solutions to an LEP problem than comprehensive understanding of how all the weapons science fits together. As discussed elsewhere in this report, often Stockpile-to-Target Sequences (STSs) and Military Characteristics (MCs) place burdens on LEPs which end up costing more in ASCI simulations and associated above-ground experiments than they are worth (at least so it seems to us), in view of changing stockpile requirements. Some specific examples will be discussed in the classified appendix. The point is that relaxing an STS or MC requirement may open the door to a relatively simple and easy-to-understand alternative in an LEP which does not require extraordinary simulations.

The second answer is that the computational burden of a sensible LEP program is so large that it certainly is an important driver in future ASCI acquisitions, both of capability and capacity. In the classified appendix, we give details of LLNL projections for total ASCI needs for the W80, including baselining, and for the W80 LEP by itself. The LEP ASCI needs are indeed considerably less than the total needs, but it is striking that if a given ASCI run for the W80 LEP is to be done in a reasonable time (defined by LLNL to be two weeks), the capability needed is multi-PF. Although the W80 LEP is only one example of the comparison of DSW ASCI requirements vs. overall ASCI requirements, this example suggests that attempts to restrict the stewardship program to DSW will not work. Not only are simulations required for DSW that are not, strictly speaking, part of the DSW computing load, the simulations needed specifically for the W80 LEP require the increase in capability the weapons labs now expect.

3.2 STS and Safety

The issue of relaxing the Stockpile-to-Target Sequence (STS) requirements in the post-cold war world was raised in the 1995 JASON report (JSR-95-320), primarily in connection with increasing the performance margins in existing stockpile weapons. The rationale was that the changed strategic situation after the demise of the Soviet Union made it timely to review whether some of those requirements still merited the costs and the burden they placed on maintaining a reliable U.S. deterrent under the new strategic circumstances.

We raise this issue again this year in evaluating the costs and burden they impose on the ASCI program as it allocates its computer assets to meet its obligations for direct stockpile support, including the life extension program. The existing STS requirements, to be sure, have a broad impact on the totality of tasks that the stockpile stewardship program must perform in order for the U.S. to maintain a safe and reliable nuclear deterrent. This is particularly evident in the analyses of performance margins of weapon primaries, and maintaining confidence that they will fully ignite the secondaries when operating in hostile environments and near the end-of-life of their boost-gas systems.

Two STS requirements that place significant demands on ASCI resources are of particular interest in this connection. One concerns the performance reliability of U.S. weapons in potential hostile environments for multiple warheads arriving nearly simultaneously at closely spaced targets (fratricide); or created by nuclear tipped interceptors from a defensive ABM system (precursor burst); or experiencing very cold environments before launch. The second requirement is for yield select options in certain types of warheads which maximize built-in flexibility in the U.S. deterrent to operate in a variety of potential scenarios. Our considerations here are not presented as a challenge to the merits of these two requirements. The DOD and the White House will judge whether the post-cold-war strategic climate and prospects merit a revision or relaxation of the STS requirements. This discussion is intended to relate and help understand the burden that they impose on ASCI in terms of the available computer capacity and capability, and the acquisition plans for new and more powerful platforms to meet the needs of the SSP.

Clearly it will be important to weigh the need for the current full suite of STS options in the nation's nuclear forces in planning a future acquisition strategy for ASCI, and indeed in scoping the entire SSP in planning for the future. These requirements do place a significant demand on the resources of the entire weapons complex above and beyond ASCI's computational power. In the new strategic environment of the post Cold War could some of the requirements for operating in hostile environments be relaxed? Given the impact of reduced performance margins on the entire certification process, could one achieve a valuable gain in the cost/benefit balance by removing extreme cold operating environments from the STS by relatively simple temperature control measures? Is the current full suite of yield select options still the most cost effective way of retaining adquate targeting flexibility for our deterrent in view of its significant burden on the SSP?

The arguments for no change in current STS requirements have to be weighed carefully in terms of their cost/benefit balance. The above numbers show that maintaining current STS requirements comes at a price when considering acquisition plans for future ASCI capacity and capability machines.

3.3 Production

The realignment of the production complex, in particular the shift of pit manufacturing from Rocky Flats to Los Alamos has resulted in the use of new processes and materials. These processes and materials must be certified, that is, shown to provide the same performance in the weapon system.

The ASCI computing capabilities are used both by the manufacturing facilities to improve and certify processes, but also by X Division to certify that the components produced will function properly in the physics package. The manufacturing facilities produce a wide range of components, including detonators, loading of neutron tube targets, shipping containers (where ASCI computing is used to evaluate damage from dropping), and, of course, pits.

3.4 The JASON "S-Matrix"

In order to organize our thoughts and queries to ASCI users, we developed a table of present and future memory and processing requirements — computing capability — for the distinct physics stages in the operation of a typical nuclear weapon. During the summer study, this table came to be known as the "S-Matrix"[1]. We asked the laboratories to estimate their current and future computing requirements for the

[1] The S in S-Matrix derives from the name of a study member who suggested this analysis approach. It also refers, figuratively, to the S-Matrix of theoretical physics.

various physics stages represented by the S-Matrix table. We found the responses to be illuminating and consistent with our own assessment of the critical physics and computing issues. In particular, the S-Matrix showed what parts of nuclear weapons codes are likely to put the greatest demands on future computers and what physical processes make these demands. From this, it is possible to infer how critical various computational capabilities will be in developing the tools to understand and predict weapons performance. In the end, there were no surprises revealed by the S-Matrix. Rather than try to display in tabular form our conclusions derived from the S-Matrix process, in what follows we comment on the important and well-known factors that underlie simulations of weapons performance.

We first remark on the recent series of calculations which led to the conclusion that the physical parameters in certain important hydrodynamic calculations converge for angular zones of size θ_c. In this series a range of angular zone sizes was used, including $16\theta_c$, $8\theta_c$, $4\theta_c$, $2\theta_c$, θ_c, $\theta_c/2$ and $\theta_c/4$, but radial zone size remained constant. Convergence of the numerics occurs when the zone aspect ratio is close to 2:1. It is well known that zones with large aspect ratios lead to numerical inaccuracies, so convergence at θ_c should not be surprising. Many estimates of future computational loads are based on the assumption of θ_c zones (although the "S-Matrix" document of July 10, 2003 provided by Charles Verdon assumes $\theta_c/2$ zones.)

This study of numerical convergence was very valuable. However, other numerical convergence issues appear to have been less well studied. For example, the number and size of radial zones is also significant. It would be appropriate to do a similar convergence study on the radial zoning (scaling the angular zones in proportion, so the ratio $r\Delta\theta/\Delta r$ remains constant, and does not exceed about 2), in order to determine how fine the radial zones must be to achieve convergence. Analogous convergence studies on other numerical parameters (for example, numerical order n in S_n transport calculations, numbers of energy groups, *etc.*) are also appropriate; we do not know to what extent this has been done.

Similar, and less well understood, issues of numerical convergence arise in many other aspect of stockpile calculations (and the "matrix"). For example, future calculations are estimated (in Verdon's document) to demand 125 times as much memory per zone and 500 times as many computations per zone per cycle to perform inline non-linear transport equations (NLTE) opacity calculations as in present calculations (this part of the calculation will then dominate the computational load). Replacement of programmed high explosive (HE) burn by reactive flow kinetics will increase the required memory 40-fold (no figures are given for the computational load). S_n transport is estimated to increase more than four-fold in memory requirement and in computational burden; this will dominate the load in both categories at this stage of the computation.

We are not close enough to the details of the calculations to say if these estimates represent an optimal allocation of computational resources. We do wish, however, to argue that it is not possible to define a calculation (or a level of computational resources) that we can state with confidence will be "sufficient" to answer the needs of stockpile stewardship. For example, HE burn and NLTE opacity reaction networks may each involve thousands of species. In how much detail must these networks be calculated in order to produce results which are "good enough" not to impair the accuracy and reliability of the final results? Will we even know when we have the answer to this question?

In principle, it might be possible to answer questions like these with a series of calculations analogous to those done to study convergence of angular zoning. However, convergence will be less obvious because there is no single parameter describing the completeness of reaction networks (much less the accuracy of constituent reaction rates) analogous to the angular zone size in hydrodynamics. In any case, the calculations have not yet been done, and may not be feasible.

Apart from finite computational resources, the accuracy and reliability of numerical calculations are limited by our understanding of the physics. Some of these difficulties are not computational at all, while others would require unimaginable computational resources to resolve. For example:

1. The equation of state of Pu is not accurately known, and no clear path to its understanding exists, even though it is, in principle, only a matter of the Dirac equation and electromagnetic interactions. Not only is the equilibrium equation of state required, but it may also be necessary to understand the kinetics of phase transitions.

2. Metals are generally polycrystalline, with grain sizes typically in the range 10–100 μ. The individual crystallites are elastically anisotropic, so that a proper calculation of shock propagation (and also subsonic flow) in such a material would require resolution of the individual crystallites. This would imply zone sizes of order 1 μ, and $\sim 10^{15}$ zones to resolve a 10 cm object.

3. When shocks reflect from material boundaries jetting and spall often result, and materials partially mix. This is the result of the growth of hydrodynamic and elastic instabilities. Mix is therefore extremely sensitive to initial conditions and instability growth parameters.

4. Solid objects undergo unstable flow and fragment when subject to explosive loads. This is also extremely sensitive to initial conditions.

In each of these examples the problem is not just a computational task which may be several orders of magnitude beyond prospective resources. It is also a lack of either basic understanding (for example, how to compute the Pu equation of state to sufficient accuracy) or unresolvable ignorance of the initial conditions (for example, the precise configuration of crystallites in a polycrystalline metal, or the initial conditions of instability growth).

The traditional approach to problems of this kind has been a combination of subgrid modeling and tunable parameters. The vast increase in computational resources promised (and already delivered) by ASCI is unlikely to change this qualitatively, although the fidelity of the subgrid models has, and will continue to, improve and the extent to which results must be adjusted with tunable parameters has, and will, decrease.

This has implications for the role of calculations, both capacity and capability, in stockpile stewardship. No foreseeable calculation can substitute for experiment. This is partly because a calculation is not a physical experiment, which may uncover unexpected aspects of the physical world; this would be true even if all of the specific uncertainties discussed above were resolved. It is also true simply because it is not known (and will not be known for the foreseeable future) how to resolve these uncertainties—the microphysics is not understood.

Consequently, the purpose of capacity calculations must be to develop qualitative understanding rather than quantitative answers (recall Feynman's remark that if you understand a phenomenon, you know what will happen without calculating it in detail); this is not directly applicable here, but does point out the importance of qualitative understanding. Once calibrated by physical experiment, this understanding makes possible the engineering judgment which has always been the basis of the nuclear weapons program. When experiments included nuclear explosions, understanding was verified directly. Now understanding must be developed from extensive series of similar calculations in which one or a small number of parameters are varied, for it is the sensitivity to such variation which much be understood.

This leaves the question of capability calculations. What is their purpose? It cannot be to replace physical experiment as an ultimate benchmark, because they are not physical experiments. It is, in part, to demonstrate computational capability, and thus to guide future code and computer development. It is also to serve as prototypes of the capacity calculations of the future, as today's capability calculations are tomorrow's capacity calculations. But because, by definition, capability calculations have turn-around times of months (rather than days), they have limited usefulness in developing the kind of qualitative understanding that must be at the heart of stockpile stewardship. The sole goal of ASCI must not be a "button to bang" end-to-end capability calculation of an entire nuclear weapon, for such calculations alone cannot provide the understanding necessary to ensure the reliability of the strategic stockpile.

4 SCIENCE: ACCOMPLISHMENTS, CREDIBILITY AND OPPORTUNITIES

4.1 Introduction: A Central Role for ASCI in SSP Science

During the course of this study, we were presented work that represents significant scientific advances in areas relevant to SSP. Some of these are discussed below and in the classified appendix. We wish to emphasize that only through better science can the ASCI codes ultimately attain the goal of predictive capability.

It will never be feasible to fully simulate the detailed behavior of the various physical stages of nuclear weapons at the single-nucleon or single-atom level. Therefore phenomenological models will always be required to account for certain "sub-grid" dynamics. This is a perfectly reasonable way to simulate nuclear weapons as long as the physical bases for the sub-grid models are understood and their ranges of validity are both understood and adhered to in the simulation. A related issue is the degree to which numerical simulation correctly models physical processes when the underlying physical laws are perfectly known. In the ASCI world, this is known as *verification.* *Validating* phenomenological models against small-scale and integrated experiments and verifying them against physically sound test problems is essential to stewardship and, we believe, is the only way toward attaining eventual *sufficiency* in required computing resources.

One can never predict when a new scientific discovery will take place. In the SSP, much effort has been expended over its formative years on the startup of ASCI and on meeting initial program milestones. With the accumulated experience and knowledge derived from this effort, we are beginning to see truly innovative scientific advances emerging from the program, some of which are described below. It is very important to maintain a certain managerial "head-room" for the most creative scientific workers in order to foster continuing scientific advances.

After describing two recent advances that have come directly out of the ASCI program, we comment more on the important issue of validation and present a JASON contribution illustrative of possible future directions. It shows how scientifically validated sub-grid models of turbulent mixing may become feasible.

"Toy" models are used in many areas of physics to illuminate complex phenomena by comparing to an admittedly over-simplified physical model, but one where relevant symmetries and dynamics are explicitly maintained. The value of such models is that they are tractable—one can find exact analytic solutions or relate parameters to relevant physical quantities. From such models, one can verify computational solutions against the analytic solution, for example. In other cases, apparently disjoint data can be organized through the toy model by means of scaling laws not possible to derive only from dimensional considerations, for example. In addition, a toy model may capture the essential qualitative behavior of a much more complex (and incalculable) physical system. Work presented to us by Hurricane and Moran (described in the next section) is an excellent example of the utility of toy models.

We wish to promote the use of toy models wherever possible in the ASCI program, both for validation and verification. In particular, we suggest that ASCI be the vehicle for all validation efforts comparing experiment with models. In this way, ASCI models are exercised against real-world problems. We further support the use of toy models to extend computational models in order to derive useful scaling laws that can describe systematics found in UGT data and that can lead to quantitative metrics for comparing simulations with experiment. This is particularly important for radiography studies which currently rely on qualitative judgments based on images. Such metrics naturally fit into SSP's Quantative Measures and Uncertainties (QMU) methodology.

4.2 Notable Scientific Advances and Their Impact on ASCI Programmatics

As we have discussed elsewhere in this report, simulations by themselves are not enough. The codes used in the simulations must be verified, to see that convergence is reached with appropriate zoning, for example. And whether the codes are fully verified or not, simulations often are inordinately time- and memory-consuming. It is tempting, when faced with complex codes and long-running simulations, for designers and simulators to spend nearly all their time dealing directly with the codes and simulations available to them rather than to step back and take a fresh look at the problem. We report here on two exceptions to this temptation, in which Omar Hurricane and Bill Moran from LLNL did step back for a new view of certain simulation problems, and in so doing provided some notable scientific advances with important implications for the ASCI program. Considered simply as science, the work of Hurricane and Moran is very nice, but beyond the implications for understanding weapons science this work promises (in Hurricane's problem) to speed up certain simulations by a very significant factor and (for Moran's problem) to provide valuable benchmarks for code verification.

We cannot discuss the details of the Hurricane problem; it is enough to say that it deals with an issue of materials failure during implosion. Over the years, when the powerful ASCI codes and machines were not available, it has become traditional to evade the problem with a phenomenological fix applied to the codes of the day. Now that simulations are so much more powerful, designers and simulators look forward to using a physics-based failure model which, while undoubtedly the most accurate tool to use for the problem, enlarges the needed computing capability by a substantial amount. What Hurricane has done is to use a scaling model invented by Neville Mott in 1947, which has proven in many applications (not just weapons science) to capture the basis physics of failure in such a simple way that it can reduce the computational load by a large factor while at the same time being faithful to the physical phenomena of failure. Hurricane's first results suggest that using the Mott model gives a good description of UGT data, so that the standard phenomenological fix may be largely or entirely unnecessary.

Mott devised his failure model in 1947, in an attempt to understand the unexpected and catastrophic failure of the hulls of 200 World War II Liberty ships, the first ships with fully-welded hulls, when subjected to the icy waters and storms of the North Atlantic. The model describes the functional dependence of various kinds of energies associated with material failure on the scale of the size of cracks which emerge and grow as a material fails. Since Mott's work, many years of careful investigation of much more complex models have verified the Mott scaling laws.

The standard for complex physics-based failure models is the Gurson model, which has evolved into many modified Gurson models. Which modifications are applied depend on the details of the problem. Gurson models are very costly to simulate because they are mathematically akin to anisotropic hydrodynamics in a medium which undergoes phase changes and discontinuous motion. Simulating a Gurson model involves the numerical solution of coupled partial differential equations. Modified Gurson models used in connection with hydrodynamics problems often result in excessive computational burdens compared to pure hydrodynamics, and have several empirical constitutive constants that often enough are unknown. Hurricane showed that simulating the Mott model of failure was orders of magnitude faster than solving the modified Gurson equations, and that it fit well with data available on failure in the context of weapons physics. The prospect lies open of replacing a phenomenological fix with simple yet accurate simulations.

Moran's work helps to verify the ability of codes to simulate various hydrodynamic problems faithfully. He has constructed a number of solutions to model hydrodynamic problems—solutions which are either analytic and exact, or which are found by the solution of ordinary differential equations rather than the partial differential equations of hydrodynamics. It is so easy and fast to solve ordinary differential equations numerically that these may be considered for all practical purposes as exact. Again, we cannot describe the details here. If we could, it would become clear that Moran has gone far beyond classic analytic solutions such as those found by Noh and Sedov, based on special cases where the hydrodynamic quantities depend on a single combination of independent variables (space, time, initial conditions). Some of his solutions, while exact, require symbolic manipulation programs such as Mathematica to do the very complicated algebra involved. While exact solutions, which typically have a high degree of symmetry, cannot be entirely realistic, Moran's solutions incorporate a surprisingly rich variety of physical phenomena against which numerical simulations can be benchmarked.

We think that the work of Hurricane and Moran are outstanding examples of thinking through a problem, rather than just simulating it. One aspect of large-scale computer systems is their ability to soak up virtually unlimited amounts of designers' and simulators' time as they struggle to meet deadlines and produce as many simulations as possible. It can happen that extensive computation can become an enemy of good science. We think that all designers ought to have the opportunity, indeed the obligation, to step away from the demands of running simulations and to consider in an unhurried way over an extended period of time how they could improve their use of high-performance computers by finding or inventing clever techniques analogous to the examples given here.

One possible mechanism would be to give designers mini-sabbaticals, perhaps one month of the year, in which they would be expected to think just about the physics of the phenomena they are simulating, and not at all about the simulations themselves. We recommend that the laboratories consider seriously what mechanism could be used to incubate physics-oriented thinking, whether the mini-sabbatical or something else.

4.3 Assessment of Scientific Credibility

A credible scientific approach to SSP requires several ingredients: sound basic science (theoretical and experimental), defensible phenomenology, experimental tests and inputs, sufficiently accurate simulation, and criteria for "good-enough" results. This report is mainly concerned with simulation, but the other ingredients are strongly coupled to simulation, and must be considered.

The SSP depends crucially on the scientific understanding of certain basic physical processes and materials properties. These can be grouped into the following topics:

1. Equation of state of weapon materials
2. Constitutive properties of weapon materials
3. Aging of weapon materials
4. Radiative cross sections
5. Nuclear reactions
6. Detonation of high explosives
7. Response of materials to shock waves, including motion, deformation, and failure
8. Interface dynamics: Spallation, instabilities, mix
9. Neutron transport
10. Radiative diffusion and radiation transport
11. Hydrodynamics of liquids, gases and plasmas
12. Hydrodynamic instabilities and turbulence
13. Dynamics of fast charged particles in plasmas
14. Effect of radiation on matter

The three weapons labs have since World War II carried out impressive research programs on all these topics, in support of their mission, and are among the world leaders in many of them.

Some of these topics are mature enough that we can have high confidence in their support of SSP: These include 4, 5, 9, 13. Others are fairly mature, but with remaining issues and even controversies: These include 1, 14. Some are immature, rapidly advancing topics with much current activity, including 2, 6, 7. Some are extremely thorny problems such as 8 and 12.

The leverage, or lack thereof, of current scientific uncertainties into the final judgements on stockpile certification is of key relevance to the SSP. In particular, LANL and LLNL have recently established criteria for Quantification of Margins and Uncertainties (QMU). QMU will provide a clear framework to evaluate these uncertainties, and is already helping to guide the program.

It is worth emphasizing that basic parameters need not be actually derived from the laws of physics; many are measurable perfectly well in the lab. If so, there is no uncertainty about the parameter. For example, the static density of plutonium is actually rather difficult to calculate theoretically (due to the complicated phase diagram) but can be measured perfectly well. The laws of physics provide the principles for isolating the basic parameters, often by arguments based on symmetry or conservation laws.

If an uncertain effect has significant leverage into the final assessment of a weapon, one may be forced to model it with a crude "fudge factor", or, as is often said, a "knob" in the code. A knob is not a basic parameter. The knob is adjusted until the code gives the correct answer, in simulating some previous measurement of a nuclear test, or other datum. The pitfall, of course, is that the knob may need to be adjusted differently for each test, and if so the code gives no real predictive power for devices different from any test. Older codes had many such knobs; new ASCI codes have fewer knobs. The newer codes therefore have, at least in principle, better predictive power than the the older ones, as well as higher resolution and 3D geometry.

The codes must simulate physics *correctly.* Experimental tests of code validity is the responsibility of the "Verification and Validation" component of the ASCI Program. The conclusion of the JASON group is that the program of such experimental tests—both large and small tests—is extremely important, and should be strengthened. Direct participation by weapons design physicists is highly desirable. To make best use of their effort, and to focus the tests on the crucial issues, the experimental tests for code validation should align with important issues of physics, materials, processes or design. Experiments at the major facilities—such as NIF, DAHRT, Z, and (for subcritical tests) NTS—can address some of the extreme conditions, and will play an important role, but small scale experiments are equally important and have the advantage in quick turnaround and rapid improvement.

Theoretical models can likewise provide important code tests. The long standing tradition of running test problems, the answer of which is known analytically, remains active, and should be strongly encouraged through invention of new analytic treatments.

To sum up, the scientific credibility of the ASCI program rests on basic science carefully applied, with appropriate control of the real uncertainties. ASCI cannot, and need not, achieve "full fidelity physics" in the foreseeable future. Rather, each phase of the simulation should provide "good enough" results to maintain confidence in the stockpile, in the sense of QMU.

4.4 Simulating Turbulent Hydrodynamics

The behavior and turbulence characteristics of hydrodynamic regions that have had sufficient time to evolve, i.e., for flows sustained for times, τ, of order of, or large compared to the outer flow time scale,

$$\tau \equiv \frac{L}{U},$$

where U is a characteristic velocity across a flow region of extent L, are characterized by the flow Reynolds number, Re, and Mach number, Ma. The Reynolds number is given by,

$$Re \equiv \frac{\rho U L}{\mu},$$

where ρ and μ the density and (shear) viscosity of the fluid. By way of example, for a region approximated as a Rayleigh-Taylor mixing layer, L can be parameterized by h, the extent of the mixed-flow region transverse to the interface, and U by $\dot{h}$, the rate of growth of h [Cook & Dimotakis 2000]. In this case, $Re \simeq \frac{\rho h \dot{h}}{\mu}$. The relevant Mach number is the *turbulence* Mach number, i.e.,

$$Ma_t \equiv \frac{u'}{a},$$

where u' is the turbulent (i.e., flow speed rms) and a is the speed of sound. A graphic representation of various flow states in the (M, Re) plane is depicted below in Figure 4-1 The figure is adapted from the recent NAS/NRC report on High-Energy Density Physics. The flow state representation and discussion below ignores coupled radiation (rad-hydro) effects that add an additional dimension and significant complexity and computational burden in such simulations. The phase space dimensionality of flow in the low-M (incompressible, $M < 0.4$) regime can be estimated from the dynamic range of the associated spatial scales. The dimensionality of low-Re flow is relatively low and can be assumed negligible in the present context. Many flows of interest here, however, are in the high-Re regime, i.e., $Re > 10^4$, and (if they have had time to develop) turbulent [Dimotakis 2000]. See Figure 4-1. Turbulent flows are characterized by a ratio of max-to-min spatial scales given by [Kolmogorov 1941]

$$\frac{L}{l} \approx Re^{3/4},$$

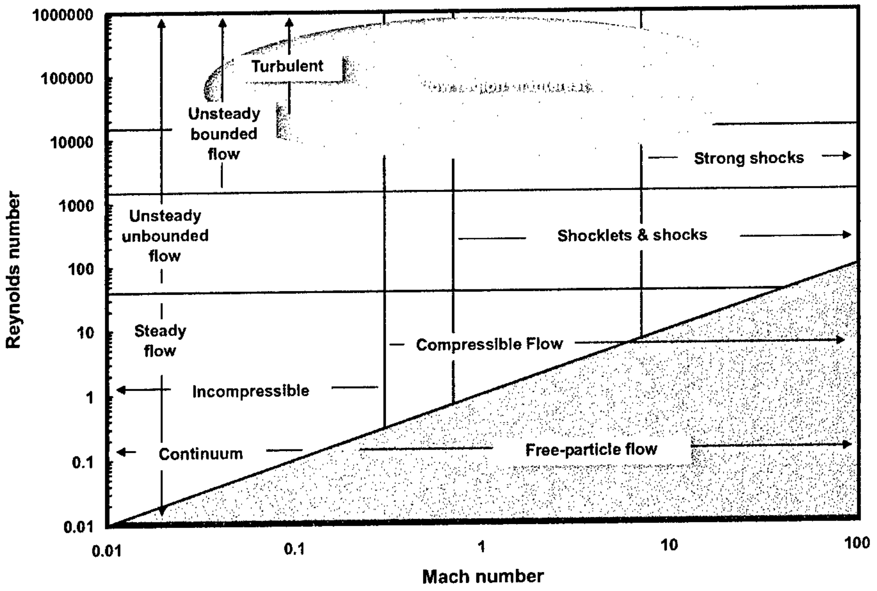


Figure 4-1: Compressible turbulent flow regimes

where l is the inner viscous (smallest) scale and the constant of proportionality is of order unity, such a region then requires (for a 3D simulation) a number of spatial-resolution elements (zones) given by,

$$N_{\text{zones}} = \left[\frac{L}{l}\right]^3 \approx Re^{9/4}.$$

The computational burden for such a simulation rises faster yet with Reynolds number, since time stepping scales with l, or,

$$t_{\text{step}} = \frac{\ell}{U} = \frac{L}{U} \times \frac{\ell}{L},$$

and therefore, the number of time steps (computational cycles) per outer-flow time scale is given by,

$$N_{\text{cycles}} = \frac{L/U}{t_{\text{step}}} = \frac{L}{l}.$$

Therefore, the computational effort required to complete a full simulation, in both time and space, scaled by the number of cycles times the number of zones, is given by,

$$N_{\text{cycles}} N_{\text{zones}} = \left[\frac{L}{l}\right]^4 \approx Re^3.$$

At present, uniform-density flows can be fully simulated (Direct Numerical Simulation – DNS) to a maximum Reynolds numbers of order $Re_{\max} \approx 10^4$, in ASCI-class machines. Variable-density flows with body forces (acceleration), of interest here, can be simulated to Reynolds numbers, roughly, 5 times smaller.[2] As can be appreciated, full 3D simulation (DNS) of such flows over the Reynolds number regimes of interest highlighted in the figure is beyond reach. The situation is also not going to significantly improve in the foreseeable future, as an increase in Re of a factor of two, for example, increases the computational burden by (nearly) an order of magnitude.

Fortunately, Reynolds number effects on mixing and other phenomena are weak, provided Re is sufficiently high, in particular, if $Re > Re_{\mathrm{trans}} \simeq 1 - 2 \times 10^4$, the threshold of the so-called *mixing transition* [Dimotakis 2000]. While it may prove infeasible to devise reliable sub-grid-scale (SGS) models, even for high-Re flows, it may be *adequate* to perform a surrogate viscous simulation at a Reynolds number comparable to, or only slightly higher than Re_{trans}. Accepting this, we have a computational threshold for the required spatial dynamic range, per L-size turbulent-flow "cube", given by,

$$N_{\mathrm{zones}} = \left[\frac{L}{l}\right]^3 \approx Re^{9/4} \approx (2 \times 10^4)^{9/4} \simeq 5 \times 10^9,$$

and a computational burden to compute an outer-flow turnover time of a turbulent cube given by,

$$N_{\mathrm{cycles}} N_{\mathrm{zones}} = \left[\frac{L}{l}\right]^4 = Re^3 \approx 10^{13}.$$

Such simulations can effectively serve as *validation* runs for hydrodynamic mix models in environments beyond experimental reach that could be used in coarser Large-Eddy Simulations (LES) that rely on SGS modeling.

The computational scaling numbers above apply to incompressible flows, i.e., flows at low turbulence Mach numbers. Compressible turbulence poses additional special challenges by introducing yet smaller length scales associated with shock thickness (scaled by the mean free path). It also changes the character of the evolution equations from elliptical to hyperbolic, or, worse, a mixture of locally elliptical and hyperbolic regions. This flow regime renders high-order-accurate compressible-flow simulations, i.e., simulations whose error decreases at a high power of the decreasing resolution spatial scale, particularly problematical.

To illustrate the difficulty, high-order accuracy in incompressible flow simulations is achieved by computing spatial derivatives, for example, either spectrally (multiplication by $\mathbf{k}$ in Fourier space and inverse transformation, or equivalent), or through

[2] In incompressible, variable-density (low-M) flow, with diffusion, the velocity field is not divergence-free. The pressure then becomes a dynamic variable and not just an effective Lagrange multiplier to ensure divergence-free incompressible (uniform-density) flow. Its solution requires inverting the Pressure-Poisson Equation (PPE) at every time step with a spatially variable coefficient $(1/\rho)$.

finite-difference schemes based on multi-point stencils straddling the point where the derivative is evaluated.[3] Most finite-difference schemes used in compressible-flow simulations attempt to mitigate this difficulty by considering asymmetric stencils and locally low-order schemes that inject asymmetry and add to numerical dissipation wherever high gradients arise, including regions with no shocks. In such flows, information fundamentally travels along well-defined directions on particular space-time manifolds. Some methods explicitly consistent with this requirement do exist, but are, at least at present, also low-order in the vicinity of shocks. Spectral methods generally fail; spectral representation of near-discontinuous field quantities in the vicinity of shocks leads to serious spurious disturbances (Gibbs phenomenon). Schemes that attempt to diagnose and filter such local effects are presently under development but are not sufficiently robust, as yet, for use in production codes. Since shocks process fluid, low-order schemes in the vicinity of shocks generally leave a trail of low-order errors in the flow, lowering simulation order throughout. This general difficulty renders computational convergence with respect to grid resolution slow, if not an open question at present, regardless of numerical scheme. A promising way to minimize such errors is through the use of Adaptive Mesh Refinement (AMR) schemes that can limit the effects further. The issue, however, remains a matter of current research with verification and especially validation, also difficult. Exploration of these phenomena will require complex and high dynamic range simulations that specifically target these effects, if errors and convergence in compressible turbulent regions are to be quantified.

A specific example—the interaction of shock waves with density discontinuity interfaces and the subsequent generation of subsonic mixing flows in addition to compressible pressure balance flows—helps illustrate these issues. In particular, note that there is no characteristic spatial scale associated with an interface-shock interaction. Instead, the generalization of the Rankine-Hugoniot jump conditions to oblique shocks shows that passage of a shock produces a tangential velocity jump $\Delta v \approx 3V_{\text{shock}}\theta$ at the contact discontinuity where θ denotes the angle between the shock normal and interface normal. It is usually quite small, at least for the first shock. The velocity jump is a vorticity sheet which is subject to Kelvin-Helmholtz instabilities of arbitrarily high wave number k and growth rate $\gamma \approx k\Delta v$, which leads to a layer K-H turbulence. What the shock passage does is instantly transform the interface from neutrally stable to hydrodynamically unstable. Dimensional analysis arguments indicate that the width W of the layer with K-H turbulence increases with time according to $W \propto \Delta vt$. Generalizing this example, one concludes that the shock processed material initially at rest and deposited vorticity in it. Vorticity lines are conserved in the plasma and their self-consistent advection is the source for subsonic "mixing flows" which are first created at the shortest scales — just opposite from standard arguments

[3] As stencil width increases, finite-difference schemes potentially (can be designed to) converge to spectrally accurate derivative calculation.

from geophysics, etc. which suppose large-scale flows that cascade to smaller-scale flows. We also note that an oblique shock generates a velocity jump regardless of the sign of the density jump at the interface. A second shock encounters a turbulently mixed layer with density interface normals lying in all directions and θ no longer is small. Thus, passage of the second shock deposits vorticity sheets with sonic-level velocity jumps, the mixing flows approach sonic values, and growth of the turbulent layer is correspondingly fast.

What are the implications for constructing sub-grid-scale models of turbulence? Two- and three dimensional codes indicate that most mixing occurs as a result of incompressible flows , whose source is the vorticity. Therefore, one should retain a full 2D calculation of vorticity sources, the vorticity-induced velocity as well as the large-scale part of three dimensional velocities. A model is needed to capture fine scale vortex stretching and other variations in three dimensions. An artifically increased viscosity can be invoked to limit the range of spatial scales, with corresponding heat sources to ensure conservation of energy. It would be interesting to develop a code based on $\nabla \cdot v$ and two components of vorticity instead of three velocity components.

The computational issues set forth above are common to many systems where the dominant force balance lies primarily in one-dimension and must be rapidly fulfilled. For inertial fusion and nuclear weapons computations a 1D, spherically—symmetric code gets the time-scale, compressive radial velocities, shock positions, densities and temperatures reasonably accurately. But, these codes lack the degrees of freedom to needed model thermal convection flows. There are similar examples in astrophysics, such as solar seismology, as well as in atmospheric circulation, and ocean internal waves. Thus the mainstream of computation has for sometime used two-dimensional codes which permit the calculation of vorticity flows as well as a modest renormalization of the 1D results. A key issue is: Can the radial force balance problem be separated and solved independently with only small adjustments needed to accommodate the longer time thermal convection and mixing flows which are driven by the creation of vorticity? To our knowledge, separation of time scales has been accomplished only for static force balance problems, such as radial force balance of a star and hydrostatic atmospheres, and not for dynamic situations. The dynamic problem would be quite challenging and would involve strongly time-dependent boundary values on elliptic operators.

5 COMPUTER SCIENCE ISSUES

5.1 Introduction

In this section we examine the computer and computational science issues associated with both high performance parallel architecture. We begin by providing an overview of parallel computer architecture including an assessment of current single processor node and interconnect architecture. A taxonomy of the various approaches to combining nodes and interconnect is then provided. This is followed by an assessment of the basic economics associated with processor and network design.

With this introduction we then provide rough estimates of the capability and capacity requirements for ASCI applications. The performance of ASCI kernels is then described with an assessment of how such kernels might perform on recent vector architectures such as the Earth Simulator. We then provide a set of analyses and recommendations on single processor performance, emerging parallel architectures, software engineering for ASCI applications and conclude with a discussion of management and procurement issues as they relate to present and future ASCI platforms.

5.2 Overview of Parallel Computer Architecture

All high-end scientific computers are constructed by connecting a set of *nodes* containing processors and memory using an *interconnection network* as shown in Figure 5-1. The performance of applications with high locality is largely determined by the architecture of the individual nodes. Applications that require significant amounts of global communication are often limited by the interconnection network.

5.2.1 Processing Node

Each node consists of one or more microprocessors, memory, and an interface to the interconnection network as shown in Figure 5-2. The microprocessor chip itself contains the processor as well as the first two levels of cache memory (L1 and

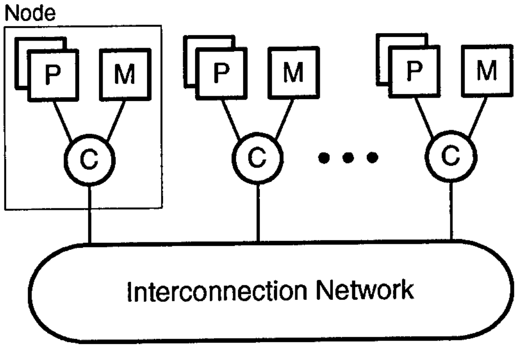


Figure 5-1: A parallel computer consists of a set of *nodes* each containing one or more processors, memory, and a communications interface connected by an interconnection network.

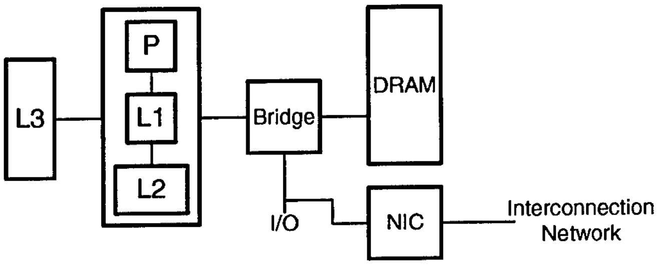


Figure 5-2: Each processing node contains one or more processors, memory, and a NIC to interface to the interconnection network. The processor includes a cache memory hierarchy with up to three levels (L1, L2, L3).

L2). A third level of cache memory (L3) is included external to the processor chip. The processor is connected to the memory and I/O system via a *bridge* chip. The network interface may be connected to either the memory system or to the I/O bus as discussed below.

While processors are typically rated by their peak arithmetic (FLOPS) rate, this number is not particularly meaningful as it is not strongly correlated to either cost or application performance. Floating-point units are very inexpensive. A fully-pipelined floating-point unit takes less than 1 mm^2 of chip area, or less than 0.5% of a $200 chip in an 0.13 μm (drawn gate length) CMOS process. With a 1 GHz clock rate, this gives a raw cost of FLOPS of about 1 GF/dollar, and this number is decreasing exponentially with time. Because arithmetic units are very inexpensive, they are overprovisioned in modern microprocessors making some other resource (usually the memory system) the bottleneck. Hence application performance is not very sensitive to peak FLOPS.

With arithmetic units overprovisioned, application performance is most strongly correlated with the characteristics of the processor's storage hierarchy (memory system). The memory system is also the area in which custom-built scientific processors differ the most from their "commodity" counterparts.

Modern microprocessors have storage hierarchies based on multiple (usually three) levels of cache memory. A cache is a small, fast memory that holds recently referenced data. A load operation first checks the cache for requested data before reading the data from main memory. Because it is small, the cache can be realized with much lower latency and higher bandwidth than the large main memory. Hence, an access that hits (is found) in the cache is completed more quickly than one that must reference main memory. A cache small enough to be accessed in just a few cycles (say 2 K words or 16KB) is too small to hold a large enough set of data to keep cache misses at an acceptable rate. Hence a second, larger but slower L2 cache is often used between the first (L1) cache and main memory. L2 is searched for data that is not found in L1 before resorting to a main memory access. Any number of caches can be used in series with each level larger and slower than the preceeding level. Most contemporary high-end "commodity" microprocessors use three levels of caches.

Table 5.1 summarizes the characteristics of a typical high-end "commodity" microprocessor's storage hierarchy.[4] For consistency data is expressed in units of 64 bit (8 Byte) words (W) and time in units of cycles (cy) (a typical cycle is between 0.5ns and 1ns). For each level of the storage hierarchy, starting with the registers and ending with off-node memory, the table shows the bandwidth provided by that level (W/cy), the latency (elapsed time required to respond to a reference) (cy), and the granularity of references (W).

Table 5.1: Storage bandwidth hierarchy of a typical high-end workstation or server microprocessor.

Level	BW (W/cy)	Latency (cy)	Capacity (W)	Granularity (W)
Registers	12	1	32	1
L1 Cache	2	3	2K	1
L2 Cache	1	8	16K	16
L3 Cache	0.5	20	512K	16
DRAM	0.25	200	1G	16
Other Node	0.001- 0.05	500-10,000	1T	16-512

The memory system of a "commodity" microprocessor, such as that shown in

[4] These numbers are modeled loosely after an Intel Pentium 4 XeonMP. The last row is a property of the NIC and network and not just the microprocessor or its bridge chip.

Table 5.1 is tuned for commercial applications, such as database systems. Such a memory system has several deficiencies when running scientific applications that exhibit considerably different access patterns. One issue is inadequate main memory bandwidth. The processor described in Table 5.1 can perform two floating-point operations per cycle but can only read a word from main memory every four cycles for a ratio of 0.12 W/FLOP. In contrast, a custom processor like those used in the Earth Simulator or Cray X-1 typically has a ratio of 0.5 W/FLOP or higher.

For scientific codes that perform random single-word memory references, e.g., gathers and scatters for sparse matrix operations, the bandwidth problem is multiplied by the large granularity of cache accesses. Each cache entry is a *line* of L words. Line sizes of commodity processors vary from L =8 to 64 words with 16 words being typical. This organization amortizes the overhead of providing cache tags for each entry, simplifies memory control, and exploits the high spatial locality of most commercial applications. However, for applications without spatial locality, caching in granularity of lines reduces the effective bandwidth by the line size L. Each time a single word is referenced at random, the entire line of 16 words containing this word is read and loaded in the cache. Similarly each time a word is written, the line containing the word is first read into the cache and then the word is written in the cache. Some time later, the entire line is written back to memory. For $L = 16$, for example, the random access bandwidth to arithmetic ratio of our example processor is reduced from 0.012 W/FLOP to 0.0078 W/FLOP. In contrast, custom processors can typically sustain their 0.5 W/FLOP on random accesses, an improvement of nearly two orders of magnitude.

The limited number of simultaneous outstanding references supported by commidity processors also limits available memory bandwidth. Most commodity processors allow at most 2–32 cache misses to occur (8 is typical) before the processor stalls. This limited amount of memory parallelism is unable to cover the large and increasing latency to main memory. With a latency of 200 cycles, for example and 8 outstanding references, the maximum rate of single-word random memory references is limited to 0.04 W/cy or 0.02 W/FLOP. It takes at least 400 outstanding references to sustain 0.5 W/FLOP with 200 cycle latency. Custom processors achieve this latency tolerance by using vector loads and stores (gathers and scatters) to launch large numbers of independent memory references. Commodity processors are not optimized for such random memory references because commercial applications achieve good cache hit rates and because they exhibit high spatial locality, making a large fraction of the 16 word cache line references on each cache miss useful.

Some custom processors, such as the NEC Earth Simulator and the Cray X-1, support vector instructions. A vector instruction performs a given operation such as load or multiply on a vector of numbers rather than on a single number. Vector machines have the advantage of reducing instruction overhead for applications that can be vectorized to use such instructions. More importantly, these machines can use vector load instructions to generate large numbers of simultaneously outstanding memory references to hide main memory latency. They have the disadvantage that the codes must be explicity vectorized in order to benefit from the vector instructions.

5.2.2 Interconnection Network

The individual processing nodes communicate via an interconnection network that can be characterized by its per-node bandwidth and its end-to-end latency. Often, the performance of the interconnection network is limited more by the network interface (often called a NIC or network interface chip) than by the network fabric itself. The bottom row of Table 5.1 shows the characteristics of a typical interconnection network with the numbers in italics reflecting the network of a shared memory system. The other numbers reflect a typical message-passing system.

The interconnection network should not be confused with a local area network (LAN), e.g., Ethernet, or a wide-area network (WAN). A well-designed interconnection network offers much higher bandwidth and much lower latency and overhead than a LAN or a WAN by exploiting the efficiencies of operating within a machine room.

It is quite feasible to provide flat memory bandwidth across thousands of nodes. Such high bandwidth is expensive, however, and not required by most applications. Thus most machines, even custom-built, have a bandwidth taper with per-node network bandwidth less than local main memory bandwidth and with the per-node bandwidth often decreasing with the distance of the communication. The Earth Simulator for example has a global bandwidth of 0.025 W/FLOP, a factor of 20 less than the local memory bandwidth. In contrast, ASCI Q has a global bandwidth of 0.004 W/FLOP.

A well-designed interconnection network connecting up to several thousand nodes in a machine room can have a latency that is a small multiple of the local memory access time—in the range of 100 to 500ns (100 to 1000 cy). A latency in this range enables full bandwidth to be sustained with relatively small granularity (1 word) transfers if a sufficient number ($\sim$50) of simultaneous transfers can be supported.

The processing nodes may interface to the network via the memory system or the I/O system. With a memory system interface, processor loads and stores to certain addresses are converted to network messages by hardware and injected into the network. Hardware at the remote node handles the memory reference and returns the requested data. With a memory system interface, the interface latency is very small (a few cycles), and there is no load on the processor (occupancy) per remote reference. Hence very short (single word) remote references can be supported efficiently. On some systems (such as the SGI O-2000 and O-3000) a cache coherence protocol is implemented by this hardware allowing local caching of remote data. On other systems (such as the T3E and X-1) data can only be cached on its *home* node.

In the case where the network interface is attached to the I/O system, a processing node uses software to send messages to other nodes and receive messages from other nodes. Because software is involved in handling each message there is a substantial increase in both latency and occupancy. In a typical system, a message send takes about 10 μs (10,000-20,000 cy) with a processor occupancy of about 5 μs (5,000 to 10,000 cy) split between the sending and receiving processors. The high latency and occupancy of a message-based network interface makes the actual latency of the interconnection network largely irrelevant. These factors also require that messages be large (hundreds of words) to amortize the latency and occupancy.

5.2.3 Taxonomy of High-End Scientific Computers

High-end scientific computers can be classified into four categories based on the processors they are constructed from and their basic system design.

Custom: Custom systems are built from the ground-up for scientific computing. They use custom processors built specifically for scientific computing and have memory and I/O systems specialized for scientific applications. These systems are characterized by high local memory bandwidth (typically 0.5 W/FLOP), good performance on random (gather/scatter) memory references, the ability to tolerate memory latency by supporting a large number of outstanding memory references, and a interconnection network supporting inter-node memory references. Such systems typically sustain a large fraction (50%) of peak performance on many demanding applications. Because these systems are built in low volumes, custom systems are expensive in terms of dollars/peak FLOPS. However, they are typically more cost effective than cluster-based machines in terms of dollars/random memory bandwidth, and for some bandwidth-dominated applications in terms of dollars/sustained FLOPS.

Commodity-Cluster: Systems built by combining inexpensive off-the-shelf workstations (e.g., based on Pentium 4 Xeon processors) using a third-party switch (e.g., Myrinet or Quadrics) interfaced as an I/O device. Because they are assembled from mass-produced components, commodity cluster systems offer the lowest-cost in terms of dollars/peak FLOPS. However, because the inexpensive workstation processors have lower performance memory systems, single-node performance on scientific applications suffers. Such machines often sustain only 0.5% to 10% of peak FLOPS on scientific applications, even on just a single node. The limited performance of the interconnect can further reduce peak performance on communication-intensive applications.

SMP-Cluster: Systems built by combining symmetric multi-processor (SMP) server machines with an interconnection network accessed as an I/O device. These systems are like the commodity-cluster systems but use more costly commercial server building blocks. A typical SMP node connects 4–16 server microprocessors (e.g., IBM Power 4 or Intel Itanium2) in a locally shared-memory configuration. Such a node has a memory system that is somewhat more capable than that of a commodity-cluster machine, but, because it is tuned for commercial workloads, not as well matched to scientific applications as custom machines. They also tend to sustain only 0.5% to 10% peak FLOPS on scientific applications. Because SMP servers are significantly more expensive per processor than commodity workstations, SMP-cluster machines are more costly (about 5×) than commodity-cluster machines in terms of dollars/peak FLOPS.

Hybrid: Hybrid systems are built using off-the-shelf high-end CPUs in combination with a chipset and system design specifically built for scientific computing. They are *hybrids* in the sense that they combine a commodity processor with a custom system. Examples include Red Storm that combines an AMD "SledgeHammer" processor with a Cray-designed system and the Cray T3E that combined a DEC Alpha processor with a custom system design. A hybrid machine offers much of the performance of a custom machine at a cost comparable to an SMP-cluster machine. Because they use an off-the-shelf server microprocessor that is produced in larger volume than a custom processor, their cost is reduced compared to a custom machine. Using an off-the-shelf processor also enables hybrid machines to leverage existing system software and compilers built for this processor. The bridge chip of a hybrid machine (Figure 5-2) can incorporate features, such as the E-registers of the T3E, that extend the memory system of the processor to allow large numbers of outstanding memory references and to perform scatters and gathers. Such features can allow a hybrid machine to match the main memory bandwidth, and random access bandwidth, of 0.5W/FLOP of a custom machine. Because of the custom system design, a hybrid machine is slightly more expensive than an SMP-cluster machine in terms of dollars/peak FLOPS. However, because it leverages an off-the-shelf processor, a hybrid system is usually the most cost effective in terms of dollars/random memory BW and for many applications in terms of dollars/sustained FLOPS.

If the goal is to maximize peak FLOPS per dollar, the system of choice is the commodity-cluster machine. This machine also give the best cost-performance on applications that cache well and hence are not sensitive to bandwidth (main memory bandwidth, random memory bandwidth, or inter-node bandwidth).

For applications where performance is determined more by bandwidth rather than peak FLOPS, hybrid machines offer the most cost-effective solution. While several times more expensive than the commodity cluster machine in terms of dollars/peak FLOPS, they offer significantly better dollars/bandwidth than any of the clustered machines. Also, because of their off-the-shelf processors, they offer better dollars/bandwidth than full-custom machines.

Custom machines are only justified for applications where the custom processor gives a large improvement in application performance. The bandwidth advantage of these systems can be more economically acquired in a hybrid machine.

SMP cluster machines are at a disadvantage for both very local applications, where commodity-cluster machines offer a cost advantage, and bandwith-dominated applications, where hybrid machines offer more cost effective bandwidth. One pays a large premium for an SMP node with a memory system tuned for commercial applications which has limited advantage on scientific codes.

5.2.4 Economics of Computer Hardware

The cost of parallel computer systems can be divided into processor cost and system cost. Processor cost is driven by unit volumes and non-recurring design costs as illustrated in Table 5.2.[5] The annual unit volumes of commodity processors (PC processors, like a Pentium 4), server processors (used in SMPs, like a Power 4 or Itanium2), and custom processors (such as used in an NEC SX-6 or Cray X-1) vary by four orders of magnitude.[6] Over 200 million PCs are shipped a year, and a given

[5] The numbers in this table are estimates based on data published in Microprocessor Report and other sources and, while not exact, illustrate the important qualitative factors.

[6] This analysis may overestimate non-recurring unit costs by using annual unit volumes rather than lifetime unit volumes to amortize the total non-recurring costs. A typical processor has a lifetime of two to three years and may have lifetime volumes 1.5 to 2 times the annual volume shown here.

processor model may account for a quarter of this volume, or 50M units. A few million server processors ship each year with one processor model reaching volumes as high as 500K—some models ship in much lower volumes. A typical custom processor ships approximately 5K units per year.

Table 5.2: Recurring and non-recurring costs of processor development.

Type	Volume	Non-Recurring	Recurring	Cost	Price
Commodity	50M	$100M	$50	$52	$200
Server	500K	$50M	$150	$250	$1,000
Custom	5K	$20M	$200	$4,200	N/A

Because commodity processors are manufactured in high volume, a great deal of engineering effort is put into raising the clock rate of these processors (Pentium 4s have clock rates as high as 3 GHz) and reducing manufacturing cost (by reducing die area and increasing yields). This engineering effort results in a total development cost on the order of $100M. This assumes the use of an existing instruction set and hence existing software. The $100M is for hardware only. Even with this high development cost, the total cost of a PC processor is dominated by recurring (wafer, test, and package) costs because of the high volumes. Pricing is a multiple of this cost. MPR (Microprocessor Report) reports that the price of PC processors ranges from $72 for a Celeron 2100 to $163 for a Pentium 4-2400 to $417 for a Pentium 4-3000/800.

Server processors, being produced in much lower volumes, are less highly tuned. They have lower clock rates (e.g., 1 GHz for an Itanium2 and 1.45 GHz for a Power 4) and are less optimized to reduce die area. As a result, the development cost of a server processor is less than for a PC processor—we estimate about half the cost, or about $50M. Again, this is a hardware only number and assumes existing software can be reused. Even with this lower development cost, the lower volume makes the per-unit cost dominated by non-recurring (design) cost. Server processors offered as merchant parts (i.e., not produced by a vertically integrated system maker) are priced many times higher than PC processors. MPR reports that the original Itanium when first shipped in 2001 ranged in price from $1,177 to $4,227 depending on cache size and speed grade.[7]

Custom processors, because they are produced in such low volumes, cannot afford the level of engineering afforded to server and commodity processors. These processors are typically designed using an ASIC (standard-cell) flow that results in lower development cost ($10M to $20M) as well as lower clock rate and larger die area. The vector processors in the Earth Simulator, for example, operate at only

[7] Prices of newer server chips are not publicly available at this time.

500MHz, and the X-1 processors operate at 800MHz. These processors achieve high performance through parallelism, not high clock rate. Even with reduced development costs, the volumes of these processors are low enough that non-recurring costs result in per-unit costs significantly greater than the price of server processors. Moreover, if these processors do not use an existing instruction set, the cost of devloping system software may be many times greater than the hardware development costs.

System cost for commodity clusters is dominated by memory DRAM memory costs about $100 per GB, so 8GB (1 GW) of memory costs $800. In comparison the processor is $200 (price), the motherboard, including chipset[8] is $100, and a 1 GB Ethernet connection is about $100 per port. Power, packaging, and I/O costs round up the cost of a commodity node to about $2K. Lower latency networks such as Quadrix provide better performance at an increased cost per node.

System costs for server, custom, and hybrid machines are somewhat higher because their system components are not produced in the large volumes that commodity chipsets and motherboards are produced. The non-processor recurring costs of these machines includes memory $800, a chipset based on several ASICs $600, and a higher-bandwidth interconnection network $500. These systems are also burdened by significant non-recurring costs that must be amortized over a much smaller number of systems. Overall system prices for these classes of machines range from $10K per processor for server machines built in moderate volumes to $25K or more per processor for custom machines built in very low volumes.

One factor that affects the type of machines available for procurement is that large manufacturers that build commercial servers, PC workstations, or both are unwilling to devote a skilled design team to a low-volume product. These companies have limited numbers of skilled design teams and the opportunity cost of directing one of these teams to design a custom or hybrid high-end scientific computer is many times larger than the non-recurring cost of such a system. These manufacturers simply cannot afford to miss a product cycle addressing a $10B to $50B market to build a machine for a $0.5B market niche. The machines these manufacturers offer to the high-end scientific markets are lightly-modified versions of their commercial database servers.

[8] PC chipsets themselves cost about $30.

5.3 Capability and Capacity Requirements

There are many simulation problems of various sizes that are needed in the stewardship program. However ASCI *capability* requirements are driven by the single most stressing of simulations, the "high resolution, full-up, asymmetric 3-D" calculation of the complete operation of a weapon from initiation through final burn.

ASCI *capacity* requirements for supercomputing are driven by the number of "findings" and similar problems that must be handled each year. Therefore capacity needs will increase as the weapons age, as life extension programs proceed and as new and different materials and components are replaced in the weapons.

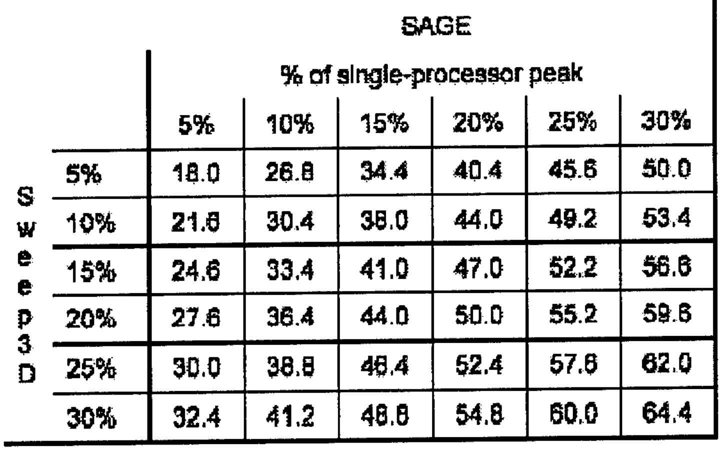
When a simulation has completed, it is important to know if it has produced a solution of sufficient accuracy. Therefore the solution will be compared to other simulations, experimental results and other tests. One of the most important tests is to increase the resolution of the simulation, repeat the simulation and compare results to the first run. If the results are sufficiently close to the first run, then the last simulation results are accepted. Otherwise the resolution is improved and these steps repeated again.

In other contexts such as simple numerical integration (e.g. numerical quadrature) the resolution is generally set to be twice as good on each iteration. For the ASCI 3D simulations, this will not be possible because capability requirements grow so fast (typically as the fourth power) as a function of resolution. The resolution can only be increased a little bit, up to the point where the available capability is exhausted. Therefore if a machine with sufficient capability over that provided by a capacity machine is not available it will be extremely difficult to validate codes that are to be used for capacity simulations. As a result, there will always be a need for at least one supercomputer in the ASCI program that has the capability to run simulations with higher resolution than the super computers used to satisfy the capacity requirements. The hardware requirements of the "capability supercomputer" need to be specified in terms of the needed resolution in the simulation.

5.4 Performance Issues for ASCI Codes

In evaluating present ASCI machines and future ASCI acquisitions, it is important to know how these machines will perform on the set of codes actually run in

Table 5.3: Percentage of Earth Simulator Single-processor Peak



		SAGE					
		% of single-processor peak					
		5%	10%	15%	20%	25%	30%
Sweep3D	5%	18.0	26.8	34.4	40.4	45.6	50.0
	10%	21.6	30.4	38.0	44.0	49.2	53.4
	15%	24.6	33.4	41.0	47.0	52.2	56.6
	20%	27.6	36.4	44.0	50.0	55.2	59.6
	25%	30.0	38.8	46.4	52.4	57.6	62.0
	30%	32.4	41.2	48.8	54.8	60.0	64.4

ASCI applications. This information is not contained in the "Top 500" ratings which are so widely publicized as that tests the performance on an idealized set of LINPACK problems which are not, typically, the jobs ASCI program are running.

Harvey Wasserman and colleagues at LANL have performed a set of evaluations of *run time to solution* metrics on a mix of ASCI codes, SAGE and Sweep3D, on existing ASCI machines and on the Earth Simulator. The former is based on hands-on calculations while the latter uses information on a single Earth Simulator processor running SAGE. SAGE is estimated to achieve $\leq 5\%$ of single-processor peak performance on one NEC SX-6 which is based on an Earth Simulator node with reduced memory performance. Sweep3D does not currently vectorize; it is estimated also to achieve $\leq 5\%$ of peak performance on the SX-6. In their report comparing the Earth Simulator and ASCI machine performance on ASCI problems they are careful to investigate a range of percentage of peak performance on Earth Simulator processors so they give the benefit of the doubt to the Earth Simulator machine.

A brief description of SAGE and Sweep3D is given in the LANL report by Kerbyson, Hoise, and Wasserman (LAUR# 03-1263). We do not repeat it here. These authors consider a mixture of 40% SAGE and 60% Sweep3D as representative of ASCI calculations. They then evaluate the *peak* Tera-Flop performance that an ASCI Q machine (using Alpha EV68 nodes) would have to achieve to match the performance of the Earth Simulator as a function of the percentage of single processor peak performance achieved on the Earth Simulator. These results are given in this table.

Using the estimated performance of SAGE and Sweep3D on a single Earth Simulator node, we see that an ASCI Q machine with 18 TF peak performance would be sufficient to equal the Earth Simulator in terms of *run time to solution.* This should be compared to the planned 30 TF performance of the original ASCI Q, though for budgetary reasons only 20 TF have been purchased.

If one recalls that Purple C is expected to achieve 100 TF, then the performance of this mix of ASCI codes will be much better than if they were run on the Earth Simulator. Of course, one can imagine that the actual programs for these codes could be tuned up for the Earth Simulator making the present comparison look less wonderful, but all in all it appears that the ASCI machines running ASCI code mixes compete very favorably with the Earth Simulator. Recalling that the cost is about 1/3 of the Earth Simulator (no operations and maintenance costs are estimated for either), it would appear that ASCI has chosen a good path for its machines.

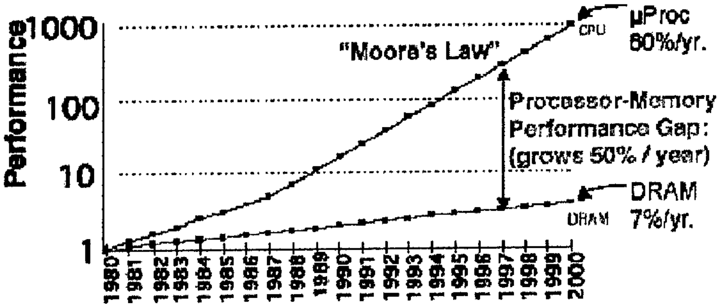
In addition to this direct comparison of commodity computers against specialty computers running a mix of ASCI codes, the LANL group evaluated the match between ASCI problems and the CPUs used in the commodity machines. In their examination of single-CPU performance they noted that the superscalar, pipelined architecture is not something over which they have control nor is the development of optimized compilers for the code they write to run on parallel machines composed of these CPUs. That said, the superscalar processors issue and execute one or more operations per clock period in separate functional units. They note, in particular, that the SGI MIPS R10000 used in the Blue Mountain machine, regarded as typical of ASCI CPUs, has one integer operation, one memory operation, one floating point multiply or add and one branching operation per clock period.

However, the ASCI codes PARTISN and SAGE each have three memory references per FLOP and have one memory reference per 2.5 to 3 instructions. This suggests that were the ASCI program to work with CPU manufacturers to produce a chip better tuned to ASCI codes, it might be possible to produce at the level of the core component of the machines somewhat better performance. Perhaps one could work with these companies to tune just those aspects of the CPUs which are a mismatch with ASCI codes and not necessarily rework existing chips into totally custom chips. If this were possible, and no discussion was given of the difficulty or cost of manufacturers doing this, there would still be an issue of the communications speed among processors, but that goes beyond the CPUs themselves.

ASCI Processors and ASCI Code

Wasserman and colleagues note, as shown in Figure 5-3, that the speed of memory has increased some 7%/year over the past two decades while microprocessor speeds have increased approximately 60%/year. So even if one were to redo the mix of instructions on the chips, the slowness of the memory could still be a bottleneck in

Figure 5-3: Processor-DRAM Gap (Latency)



getting higher performance from individual ASCI processors and local memory.

Modern superscalar processors issue and execute more than one operation per clock period in separate functional units. For example, the SGI MIPS R10000 processor (used in ASCI Blue Mountain) can issue one integer, one memory, one floating point multiply or add and one branch conditional operation per clock period. When running PARTISN there are observed to be about three memory references per FLOP and 2.4 instructions per memory reference. When running SAGE there are observed to be about three memory references per FLOP and three instructions per memory reference. Hence, typically 7–9 instructions, including three memory references, must be executed for every floating point operation, even in physics codes whose substantive output is the result of floating point arithmetic. Because of the failure of memory speeds to keep up with microprocessor speeds, memory bandwidth and latency are usually computational bottlenecks, explaining the low operating efficiencies of ASCI machines. This is not a failure of the programmers, but is intrinsic to the hardware architecture.

5.5 Analysis and Recommendations

5.5.1 Single processor performance

The ASCI computers are expensive and over-committed. Under such circumstances single processor efficiency is important. There should be an expanded program to improved the efficiency of critical codes.

Large systems are advertised as having 1 or 10 or 100 peak TF, but they don't deliver all of them to programs. To explain that, people look at how effectively important algorithm implementations use the individual processors, and get results (typically, 0.5–10% of peak) that initially seem quite disappointing.

Why does it matter?

For users the proper metric is *time to solution.* So, if one could improve performance by 25% then a 5 day run would only take 4 days, and runs of other lengths would decrease proportionally. Not only would users be enthusiastic at a higher rate of results, but more work could be done on oversubscribed machines.

There are many ways to improve performance. At the highest level, improved physics could completely eliminate some required high resolution simulations. Improved algorithms might profitably make less efficient use of the CPUs while substantially reducing the running time. More advanced algorithms may be more memory intensive, and memory bandwidth is a significant challenge in computer architecture.

Here we consider lower-level issues. When the numerical techniques are fixed, then improvements in performance come from improvements from the layout of data, or from generating better machine code, etc. Whatever the details, fixing the numerical technique pretty well determines which computations have to be done, and better performance comes from performing them more efficiently.

How bad is the inefficiency?

Both the software and the CPU chips are complex so it is difficult to give a simple answer. One measure that follows naturally from describing machines in terms of peak floating point operations (currently TF) would be the the ratio of the floating point operations the program sees divided by how many the CPU is capable of in the same time. The number varies for different algorithms and physics, but numbers from 0.5% to 10% seem typical.

Such low rates seem pathetic, but we were told that similarly low rates are seen for important commercial processing tasks. If the numbers are the same, why should ASCI care if the big commercial customers don't?

The answer is that the commercial customers manage to get all the computing done that they need. The big web sites, or the airline reservation systems, deal with essentially all the requests fast enough. The techniques they use, like buying more machines, satisfy their computing needs. The scale of the ASCI problem makes it difficult, not impossible, to simply buy more computers. For example, very large numbers of processors present a serious relability problem.

The ASCI machines are oversubscribed, probably by more than a factor of 2. Jobs get to run by some combination of administrative fiat and management priority-setting. In this environment even modest improvements in performance are worth a lot.

What is the source of the problem?

Effective use of CPUs is inhibited by the enormous speed mismatch between the cycle time of the arithmetic unit on the chip and the time it takes to get arguments out of memory and results back to memory. In modern chips the ratio can range from 90 to 200, perhaps more (see Table 5.1). That is, from the time an operation needs a word in main memory it is 150 clock cycles (the latency) before the operation can proceed. If processors really operated directly to main memory, then they would best be described as 20 MHz processors rather than 3 GHz.

Modern chips mitigate the long memory latencies with caches, with the ability to work on several instructions at once, with pipelining, and with other architectural techniques. Even after all of this, the implementations of some important ASCI algorithms starve the processor. In the worst cases, each floating point operation needs two arguments from memory and produces one result that has to go to memory. (Dense linear algebra benchmarks get their good results because the calculations can be arranged so that results are reused as operands to nearby calculations; traffic to and from memory is much reduced.) Moving three floating point numbers between memory and CPU limits the number of floating point operations that can be performed because of the bandwidth requirements. The memory bandwidth of the G4 (similar to the processor in "White"), for instance, is about 1.2 GB/s, which is enough to support 50 million of these memory-clogging floating point operations per second, which is about 3% of the peak floating point operation rate.

Thus, the memory problem arises both from latency and from bandwidth.

What about Vectors?

Vector machines were originally introduced (nearly 30 years ago) to help with these problems. (The fact that vectors seemed to be the natural mathematical structure for expressing algorithms was a useful coincidence.)

The point of the early vector machines was pipelining. That is, once a vector operation started, the CPU knew exactly what it would be doing for quite some time. For instance, once a vector load operation, for a vector n long, started, the machine knew the next n memory addresses it needed to read data from. After some initial delay the memory subsystem could deliver one operand every cycle.

There are two points to be made in this. The first is that these systems had comparatively large memory bandwidths. The Cray machines around the time of the YMP had at least half their circuitry dedicated to controlling the memory subsystems, in order to get the bandwidth delivered. This cannot easily be duplicated today in ASCI-style machines. The second point is that the logic associated with vectors was quite simple, requiring not too many gates in the days when gate counts were a bottleneck. Indeed vector machines acquired an indirect vector load, where the vector load worked with a vector full of addresses, rather than a single address and stride.

This second mode can be simulated, with lower efficiency, on modern processors. Of course the fact is well known, and is a part of what goes into instruction scheduling by compilers. Most modern microprocessors allow 2–16 pending memory operations. A compiler that is aware of the actual number can use it to boost performance.

The Earth Simulator is reported to have memory bandwidth of 32 GB/s. All else being equal, that supports 1.3 billion memory-clogging operations per second, or about 15% of peak floating point performance. There is no immediate application of this observation to comparative performance.

What has been proposed?

Here are several ideas that have been proposed. Some are unworkable and some have promise. Included is one more, that no one briefed.

Hardware Changes If one were completely convinced by the memory arguments above, there could be a push to encourage better memory performance from the vendors of ASCI capability machines. It is not clear how much flexibility there is. Certainly minor modifications of the chips (or chip sets) could plausibly be required, for modifications that can be shown to make definite large improvements in performance. It seems unlikely that minor changes are what would be needed to fix the memory bottlenecks.

In the long run, however, work is needed on alternate architectures. This is discussed briefly later.

Better Compilers The algorithms being used seem to have outstripped the ingenuity of compiler writers. There is a perception that compilers have not improved much in the last decade. Research in compiler optimization has fallen rapidly over the last decade. Some work here (probably through academic centers) seems worthwhile, but the problem is difficult. Compilers have to produce correct code for all legal programs. Some important optimizations might require knowing things that the compiler cannot deduce from the program, or require rewriting at a more global level than is feasible.

On the other hand C++ and similar languages that lack pragmas are much harder to compile well than are Fortran-like languages. Further, programming style can impede or help the compiler. Nonetheless, we believe a modest amount of money should go into compiler work. It would be most effective to get existing academic groups to do research that addresses ASCI needs, but if that is not possible, it might be worth having an internal compiler group that works with vendors and/or the open source community.

Hand Tuning Presumably single processor efficiency could be materially improved by hand tuning. That, after all, was effective in the days of vector processors.

The arguments against this approach are fairly strong. It is labor-intensive. It may result in code that is different for different machines, which requires more testing and lessens confidence in the code. More seriously, it makes improvements (or bug fixes) very difficult. To change the program, first the hand tuning has to be undone, and then the changes put in, and then the hand tuning redone. The structure of the code before tuning may well be obscure, making the changes especially difficult. Further, the people doing the code improving and hand tuning may well be different, and so the whole thing breaks down.

Automatic Code Optimization In between better compilers and hand tuning lies a set of techniques that are characterized by writing programs that write programs. One example, ATLAS, is discussed in detail. Two others are mentioned, as is a third, simple hypothetical example.

- *ATLAS for the BLAS* The web site is *math-atlas.sourceforge.net.* The program constructs subroutines for dense linear algebra tuned to the CPU it is run on.

  To give a sense of its effectiveness (recognizing that this is not a significant ASCI problem), consider the problem of multiplying dense matrices. The benchmark code is the natural triply-nested loop in the C programing language, compiled with the Gnu C compiler with maximal optimization. The benchmark was run on a Sparc. The Solaris compiler with maximal optimization produced 30% faster code. To do this it unrolled the inner loop 4 times, and then had more space to do a better job of scheduling the instructions. It also had to generate a significant amount of code that does the right thing if the size of the matrix is not a multiple of 4. ATLAS generated code that was another 10 times faster. It rewrote the algorithm almost completely so that matrix multiply was not done as n^2 inner products, which is the benchmark code, but on matrices partitioned into 4 by 8 and 8 by 4 submatrices (plus the code for uneven boundaries). It also helped the compiler by figuring out how far apart in the instruction streams the multiplies and adds should appear. The code it generates is C code that produces one floating point result every clock tick, which is impressive.

ATLAS provides tuned versions of the BLAS, the Basic Linear Algebra Subroutines. LAPACK invokes BLAS as the basic perations in dense linear algebra. The BLAS come in sorts: vector-vector operations, vector-matrix operations, and matrix-matrix operations. Matrix multiply is of the latter sort. They also come as single and double precision for real and complex numbers. In the Solaris example it took ATLAS six hours to generate all of these. Intel-based computers running Linux take a lot less time for some reason.

ATLAS works essentially by searching through a space of alternative algorithms. It figures out how big the caches are, how fast it can get floating point operations done, whether there is a fused multiply-add, and so forth. It uses these to limit the search, but it still tries a number of alternatives and parameters.

In principle compilers could find the same optimizations, but ATLAS is creating a library, and the costly optimizations have to be done infrequently. Of course, if a program is going to run for a week, then spending an hour in the compiler to get materially better code is worthwhile, but compilers don't seem to work that way.

- *FFT, and the Superoptimizer* Here are two more examples that illustrate the breadth of applicability for programs that generate other programs.
  - Henry Massalin, Superoptimizer: a look at the smallest program, in Architectural Support for Programming Languages and Operating Systems, ACM, 1987, ISBM 0362-1340.
    In this remarkable paper the author did a large search to find a way of implementing unsigned divide in the Motorola 68000. He generated instruction sequences, evaluated them on some inputs, and when they didn't work properly went on. The search was clever, but in essence he tried a vast number of instruction sequences until he found the best one. It was hard to imagine human beings finding the one he found, and compilers were using worse ones.
  - At *www.fftw.org* is machine-generated code for doing FFTs. The code is generated by an optimizer written in the CAML programming lan-

guage. For FFTs whose size is a power of 2 there's nothing special, but they generated a good program for a size 101 FFT.

- *Compiler Flags* These same techniques can be applied much more simply to the chore of finding which compiler flags to use, and in suitable circumstances could automatically decide which are the best compiler flags. Modern compilers may have dozens of flags, and it is never clear from the documentation which provide the best performance. But it is just a matter of trying all (or all plausible) combinations and seeing which is best. The code to do this is a simple scripting task. (Some compilers can use control flow information from a previous execution of a program to do a better job of compiling. Including such flags is simple.)

  If the performance of a program, or an important piece of a program, can be determined in a minute or so, then this is computationally feasible. Otherwise one needs to make up such programs and presume that their optimal flags also apply to other programs.

- *And ASCI?* More knowledge than we have would be required to decide where best to apply such techniques to the ASCI codes. But the standard way of improving programs with performance problems, once the algorithms are chosen, is to express the computation in different ways and to lay out the data in various ways, and see what happens. Varying the implementation this way is well suited to automation. One might call these projects *ATLAS for ASCI.*

Further Advice on Architecture

The program now is using machines with roughly 10 peak teraflops, and procuring machines which may provide 100 peak TF. Describing machines in terms of peak TF is at best misleading, but it may be used to describe the relative performance of, for instance, Purple C or Q. The architecture of these machines can be roughly described as having two levels. The first is high-performance commodity microprocessors arranged in nodes that are symmetric multiprocessors. The second level consists of connecting these nodes by an interconnect that supports MPI.

This two layer arrangement may not scale well to provide the even larger machines that the program will need. Neglected so far in the program is an architecture we'll refer to as the T3E architecture (so as to avoid the loaded term *vector*). This architecture would fit between the multiprocessors and the MPI infrastructure. The T3E architecture for us means an interconnect that supports a large address space, and is capable of having many (say thousands) of outstanding memory references per node.

Depending on the state of the software one of these super-nodes would appear in different guises. With no change it would just look like a multiprocessor node that supports many more threads and a much larger address space, with the penalty that the memory would seem somewhat slower. Existing code would likely run reasonably well. As local memory is faster than the other memory, the code could be modified to use a programming idiom that takes advantage of the speed. The super-nodes would still communicate with each other using MPI.

Using vector processors in place of the commodity processors would make the job of the compilers easier, but that is an independent issue, and not central.

Thus there are several long-term approaches:

- Commerical symmetric multiprocessors interconnected by MPI (the present path)
- T3E: Commercial processors sharing one large NUMA (non-uniform memory access) address space
- Symmetric multiprocessors (with or without vectors) sharing T3E-like interconnect
- Either of the last two interconnected with MPI

There is no reason to believe that the present path will last into the indefinite future, and it is time to start exploring alternatives. Various versions of the second or third alternatives have been used very effectively at significant scale in other organizations.

5.5.2 Emerging Parallel Computer Architectures

Future SSP tasks are expect to require *Capability* machines with *sustained* performance of a PF within a decade and even greater performance in later years. Reaching a PF and beyond by simply scaling conventional cluster architectures is problematic. A conventional PF machine is expected to require over 10^5 processors leading to issues of reliability, efficiency, and manageability.

A number of emerging computer architectures have the potential to ease the process of reaching a PF. Some of these technologies, like processors in memory (PIM) and streaming, lower the bandwidth demands of applications by exploiting spatial and temporal locality. In doing so, they have the potential to both reduce cost and to realize machines that sustain a greater fraction of peak performance, particularly on bandwidth-dominated applications. A number of complementary technologies, such as electrical and optical signaling technologies, and new interconnection network organizations allow global bandwidth to be delivered more efficiently. Other emerging technologies, such as vectors, streaming, and multi-threading, scale the performance of a single processor chip, allowing a PF machine to be realized with 10^3 to 10^4 processors rather than 10^5. Each of these technologies is discussed briefly below.

It is unlikely that machines incorporating these emerging architectures, or others that have not yet been proposed, will be available for acquisition unless significant investment is made in developing these architectures—to take them from concept to pilot systems. The mainstream computer industry is unlikely to develop these architectures on their own as these architectures are primarily of use to high-end scientific machines and will have limited impact on mainstream workstation and commercial server markets.

Architectures that reduce bandwidth demands

The limited bandwidth of commodity- and SMP-cluster machines is a major factor in the low fraction of peak FLOPS that are sustained by this class of machines. There are two approaches to addressing bandwidth limitations: increase bandwidth, and reduce bandwidth demand.

Several emerging architectures have the potential to reduce demand on memory bandwidth by finding and exploiting spatial and temporal locality that is not captured by conventional cache architectures. Processor-in-memory (PIM) architectures, for example, exploit spatial locality by co-locating a processor on each memory chip. The Berkeley VIRAM is an example of an experimental PIM system. Because a very wide, high-bandwidth bus can be inexpensively realized on-chip, a PIM provides very high bandwidth to the processor's local memory. Operations that reference data that is spatially local to within the processor's local memory are not limited by bandwidth constraints.

Attempts at building PIM systems to date have run into two limitations. First, the cost per bit of DRAM memory implemented on an ASIC or processor chip is significantly (at least 10×) more expensive than on a commodity DRAM chip. This causes the memory, a major fraction of system cost, in a PIM system to be prohibitively expensive. Second, the memory capacity that can be realized on a single chip (about 512Mb or 8MW) is far less than the amount of memory needed per processor. PIM-like systems with high-bandwidth commodity memory chips packaged near a separate processor chip may be able to overcome these limitations by achieving comparable local bandwidth without the memory cost and capacity constraints of a monolithic implementation.

Stream architectures exploit temporal locality by using a compiler-managed on-chip memory, called a stream register file (SRF), to capture many data references. Stream compilers reorder a computation so that the intermediate results computed during one phase of the computation fit within the SRF and are then consumed in the next phase of the computation without ever being stored to memory. The stream processor is able to use the local memory more effectively than a cache for two reasons. First, because the SRF is logically a *register* the first reference to the SRF can be made without fetching the data (and the rest of its line) from memory. This enables an SRF to capture values with no reuse (i.e., they are written once and read once) with no memory traffic. Second, the compiler that manages the SRF knows which data in the SRF is live and which is dead.[9] Hence the compiler can load new data without displacing live data. A cache in contrast evicts both live and dead data, leading to unnecessary memory traffic—storing dead data and both storing and reloading live data.

To complement these bandwidth reducing architectures, new technologies are also emerging to provide more bandwith and to provide bandwidth more cost effectively. Both electrical and optical signaling technologies have advanced rapidly in recent years. Over the past five years, electrical signaling rates in interconnection networks have advanced from 400 Mb/s (Cray T3E) to 3.125 Gb/s and they are expected to reach 20 Gb/s by the end of the decade. Faster signaling increases the bandwidth that can be passed through a router chip, a connector, or a cable, and thus increases the bandwidth that can be realized for a given cost. Parallel optical transceivers based on VCSEL technology allow these high signalling rates to be economically transmitted over distances up to 300 meters —e.g., between cabinets in a machine room.[10]

Architectures that scale single-node performance

The performance of conventional microprocessors going forward is likely to scale largely with device speed, increasing at about 20% per year. This is a reduction from the historical scaling rate of 50% per year. That rate depended on not just faster devices, but also more instruction-level parallelism (ILP) and reducing the number of gates per pipeline stage. These later two performance factors have been largely mined out. The ILP of high-end microprocessors has not increased in the last several

[9] Live data will be referenced again while dead data has no further use.

[10] While the optical technology has a longer reach, the cost per unit bandwidth of optical signaling is about 10x that of electrical signaling.

generations (the Pentium 4 actually has lower ILP than the Pentium 3). Also, the number of gates per pipeline stage is now down to less than 10, and cannot be reduced much further.

To get to a PF of performance, or even 100 TF of *sustained* performance requires scaling today's 1 TF sustained machines by a factor of 100. By the end of the decade, we can expect a factor of 3.6 from faster conventional processors, leaving a factor of 28 to be realized by scaling the number of nodes—i.e., scaling from 10,000 nodes today to almost 300,000 nodes to realize 1 PF. Scaling to such a size is likely to present a number of efficiency, reliability, and operational issues.

An alternative to scaling the number of processors is to scale the parallelism within each processor. While instruction-level parallelism (ILP) has been largely "mined out", data parallelism (DP) and thread-level paralleism (TLP) are abundant in large SSP problems. Several alternative architectures have the potential to exploit DP and TLP to scale the performance of a single processor by a factor of 100 or more[11], allowing a PF machine to be built with no increase, or even a decrease, in the number of nodes.

Vector and stream architectures exploit data parallelism, performing 8 to 16 operations in parallel for each instruction issued. Multiplying this DP by today's levels of ILP, vector and stream processors with 32 to 128 FPUs operating at 4 GHz are feasible by the end of the decade—128 to 512 GF per processor. Advances in vector register file organization, and the integration of stream processor concepts into vector processors make this level of performance feasible. Advances in conditional vector operations have the potential to run large classes of codes, including many codes that historically have not vectorized, on such machines.

Multi-threading is a complementary approach that exploits thread-level parallelism. Used primarily for network processors, chips are available today with up to 16 8-way multithreaded processors. If such architectures are redirected toward scientific computing, multi-threaded processors with FLOPS rates comparable to vector processors are feasible by the end of the decade. Hybrid processors that combine multi-threading with vectors are also possible to achieve even higher-levels of intranode parallelism.

Either a vector/stream or a multithreaded processor can use the locality enhancing technologies described above to reduce the bandwidth demands of such a node to a reasonable level. Vector load/store instructions easily generate sufficient

[11] These machines get the same 3.6× increase in clock rate and add to it a 32× to 128× increase in paralleism per processor.

outstanding memory references to hide memory latency. Because each thread is able to issue several memory references before blocking, multithreaded processors can also hide long memory latencies while sustaining high bandwidth.

5.5.3 Software Issues

The task of creating a software system for the ASCI supercomputers that will be needed to support the stewardship program for the next several decades is a daunting one. Already it is reported that simulation codes now exceed one million lines yet a huge amount of code remains to be created. This software task is a large-scale industrial engineering effort that consumes a major part of the $700 million dollars/year budget of ASCI. The effort deserves the careful care and skillful management that would be brought to any engineering project with such a budget and a product that will be worth billions of dollars and be in use for decades.

In our review of the ASCI efforts we found a very large variation in the quality level of software engineering practice. We happened to first visit Los Alamos National Laboratory and immediately after, Sandia National Laboratory. There was a striking difference between the high quality of software engineering at Sandia as compared to Los Alamos National Laboratory. Yet Los Alamos had developed some very innovative software, and our criticism is of the software engineering process and not the resulting software.

It is clear that the ASCI program must better integrate the software engineering practices of all the laboratories in the ASCI program and not just leave each laboratory to develop whatever kind of software engineering program as strikes their fancy. At Los Alamos in particular, better ways of integrating computer science (CCS Division) with weapons program (X Division) must be found. It is also very important in this process to optimally integrate the work of each type of programmer that is needed to make the effort effective and successful:

1) The very skilled and disciplined software engineer/programmer,

2) The highly innovative (and occasionally undisciplined) scientist/programmer, and

3) The "heroic" scientist/programmer/manager who often carries (successfully) a large part of the effort almost on his own (There is an obvious risk here for the overall program).

We recommend that a single, integrated software engineering process for ASCI production software be put into place for all the laboratories and components of the ASCI program. There should be an annual joint review rating and maturity assessment of this ASCI-wide software engineering process by one of the outside software engineering assessment/certification organizations such as the Software Engineering Institute of CMU (a Federal Funded Research Development Center (FFRDC)).

5.5.4 Management and Procurement Issues

Management

In general the ASCI program has managed the high end ASCI platforms extremely well. The ASCI program has made it possible for a given laboratory to carry out large scale stockpile calculations on the capability platforms of the other laboratories via a secure high speed networking infrastructure. There has evolved over the years a culture of cooperation among the three laboratories that makes it possible to prioritize the resources based on program deliverables.

However, in normal operation, the nodes of the capability platforms at the labs have been divided up with sections reserved for each laboratory. While such an approach has the advantage of providing equitable access to the platforms for each laboratory it has the downside that capability is, in large measure, being used in a capacity mode.

Future procurements will be guided by the need to balance capacity and capability. Under such constraints, and given the finite resources available to platform acquisition, capability acquisitions will be focused on the need to achieve PF level performance. Such platforms will most likely be unique and it is conceivable that one such platform will have to serve all three laboratories. It may therefore prove necessary to site a future capability platform at one laboratory and manage it centrally for all three laboratories.

ASCI platforms are similar in terms of strong user demand, large capital cost and complex operations to other forefront scientific instruments such as high energy particle accelerators and large telescopes. Useful lessons in the productive exploitation of ASCI machines for NW science can be drawn from experience with these other facilities.

DOE has a long record of supporting and operating large accelerators. When these devices had grown in size and cost to the point they could no longer be co-located with their users on university campuses, national and international laboratories were created to build and operate them for international user teams. Typically, particle detectors used in conjunction with the host accelerator are constructed by the users, but the operations of the accelerator are not charged directly to users. Selection of particular scientific proposals for access to accelerator beam time and other resources is generally a responsibility of the host laboratory. In detail, different labs exercise this responsibility in different ways, but normally a program advisory committee (PAC) advises the laboratory director on which proposals to run or not run. Proposals are examined by the PAC for scientific importance, feasibility and ability of the proponents to carry out the experiment they propose. A laboratory's PAC generally comprises active researchers, well known in the user community. Their recommendations are almost always accepted.

The process of proposal preparation and PAC review naturally creates opportunities for communication of new scientific ideas and methods throughout the user community. We feel that similar proposal review procedures can be employed to good effect in the ASCI program. In analogy to beam time, access to "most-capable" computing is the limited scientific resource of great value to users. We are aware that protocols exist in the ASCI program to do just this, but access by users outside the host laboratory is generally limited to a certain fraction of available capability or capacity. We see no reason to arbitrarily limit access and believe that community-wide reviews of proposals for "most-capable" projects could have additional benefits in terms of increased communications and scientific competition.

Procurement

Future capability procurements should be strongly guided by metrics that will allow the program to effectively judge whether a proposed platform will effectively meet the stockpile workload. Among the three labs there is now a world class body of expertise in performance metrics for high end capability platforms. Performance metrics relevant to stockpile needs (such as for example the Purple benchmarks used in the Purple acquisition) should be uniformly applied.

Once a promising architecture is identified using stockpile computational metrics, it will be important to identify early any system-wide integration issues that may arise in the process of delivering a full configuration. Future ASCI systems will continue to be unique in terms of their size (e.g. memory, storage, and network). Inevitably, system-wide integration issues not previously encountered for other smaller installations will emerge as a result of the scale of the facility.

It will never be possible to completely mitigate these risks. However, early delivery of a prototype capability platform may allow the ASCI program to gain valuable experience prior to the full delivery. In addition to providing needed capability cycles to the program, such prototypes will allow the program to fully evaluate a promising architecture under the full requirements of stockpile simulation, something that cannot be done except in a secure computing facility. Indeed, it would be important to make such an intermediate procurement a “critical decision” point in the procurement process. In section 6 we advocate that NNSA invest in “capability exploration” platforms that have the potential to deliver a PF in the 2010 time frame. Future procurements of such platforms would benefit from a requirement of delivery of a prototype platform that can deliver usable cycles to the program while serving to identify future issues associated with deployment at scale.

6 ACQUISITION SCENARIOS

In this section we consider possible acquisition scenarios for present and future ASCI platforms. We will look at two such scenarios:

1 Effect of delay in acquisition of the Red Storm and Purple C procurements, and

2 Use of commodity cluster technology combined with capability.

In the process of presenting the second scenario it will be necessary to consider the use of commodity cluster technology and its possible role in the ASCI procurement strategy. The second scenario is only meant to be illustrative and indicates how one might balance capability and capacity requirements.

Note that the assessment which follows is based on the ASCI workload for LLNL and LANL. It does not include workload requirements coming from engineering (SNL), and so what follows is merely illustrative of how workload factors into the platform acquisition strategy and drives a balance between capacity and capability computing. Most importantly, the platform acquisition costs discussed in scenario 2 are meant to be notional and should not be seen as a recommendation for budget levels for future acquisition. Once all workload requirements are in hand a similar methodology could be employed by NNSA to arrive at appropriate budget levels which are reflective of the requirements.

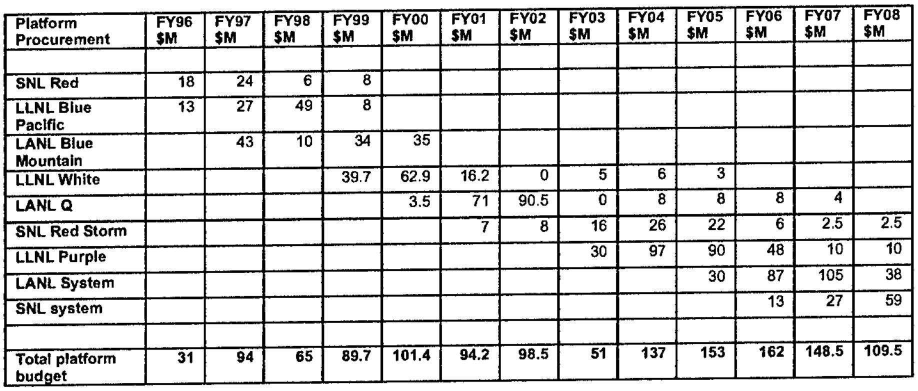
6.1 Original NNSA Procurement Strategy

As a point of reference we present in Table 6.1 the ASCI platform procurement budget as presented to us by W. Reed of NNSA.

To understand how these proposed purchases affect program requirements we include in Table 6.2 the peak capability (in TF) associated with each proposed platform as well as an aggregate measure of overall capability and capacity available to the tri-lab complex.

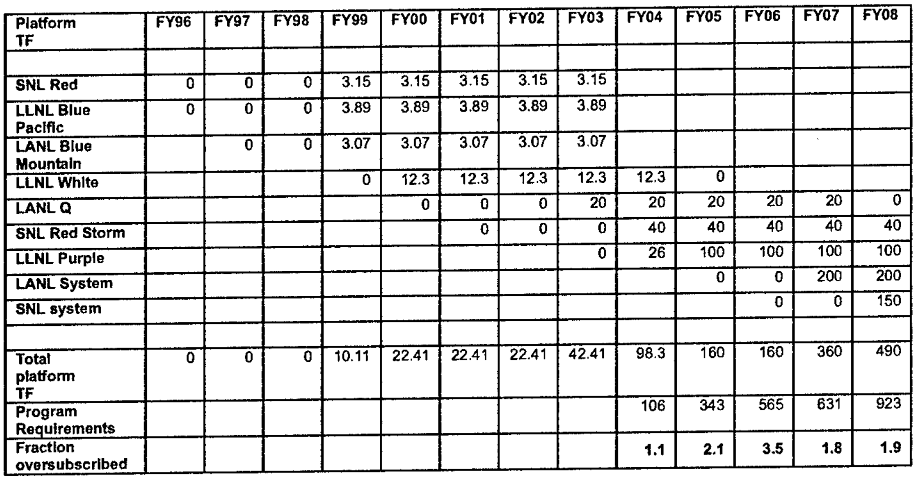
As discussed in our report, peak ratings are not generally indicative of actual capability and in the future it will be useful to rate such acquisitions in terms of

Table 6.1: ASCI Platform Procurement Budget



Platform Procurement	FY96 $M	FY97 $M	FY98 $M	FY99 $M	FY00 $M	FY01 $M	FY02 $M	FY03 $M	FY04 $M	FY05 $M	FY06 $M	FY07 $M	FY08 $M
SNL Red	18	24	6	8									
LLNL Blue Pacific	13	27	49	8									
LANL Blue Mountain		43	10	34	35								
LLNL White				39.7	62.9	16.2	0	5	6	3			
LANL Q					3.5	71	90.5	0	8	8	8	4	
SNL Red Storm						7	8	16	26	22	6	2.5	2.5
LLNL Purple								30	97	90	48	10	10
LANL System										30	87	105	38
SNL system											13	27	59
Total platform budget	31	94	65	89.7	101.4	94.2	98.5	51	137	153	162	148.5	109.5

Table 6.2: Capability and Capacity



Platform TF	FY96	FY97	FY98	FY99	FY00	FY01	FY02	FY03	FY04	FY05	FY06	FY07	FY08
SNL Red	0	0	0	3.15	3.15	3.15	3.15	3.15					
LLNL Blue Pacific	0	0	0	3.89	3.89	3.89	3.89	3.89					
LANL Blue Mountain		0	0	3.07	3.07	3.07	3.07	3.07					
LLNL White				0	12.3	12.3	12.3	12.3	12.3	0			
LANL Q					0	0	0	20	20	20	20	20	0
SNL Red Storm						0	0	0	40	40	40	40	40
LLNL Purple								0	26	100	100	100	100
LANL System										0	0	200	200
SNL system											0	0	150
Total platform TF	0	0	0	10.11	22.41	22.41	22.41	42.41	98.3	160	160	360	490
Program Requirements									106	343	565	631	923
Fraction oversubscribed									**1.1**	**2.1**	**3.5**	**1.8**	**1.9**

metrics that are more reflective of program requirements. We assume in Table 6.2 that each platform, once deployed, has a utilization period of five years and then that platform is retired.

An assessment of future SSP requirements was presented by J. Peery and J. Rathkopf for LANL/LLNL. These estimates were obtained through extensive discussion with the designer community and take into account important program deliverables such as the W76 LEP as well as the need to baseline the ASCI codes for the stockpile systems that are the responsibility of the respective laboratories. These estimates (in terms of required platform peak TF) are expressed in the row labeled "Program requirements". As the ASCI codes are increasingly deployed for stockpile assessment and certification activities, a very clear increase is seen in these requirements. It is of course difficult to assess the accuracy of these requirements as they represent in many cases conservative estimates of what will be required to perform the required computational tasks associated with a program deliverable such as an LEP or SFI.

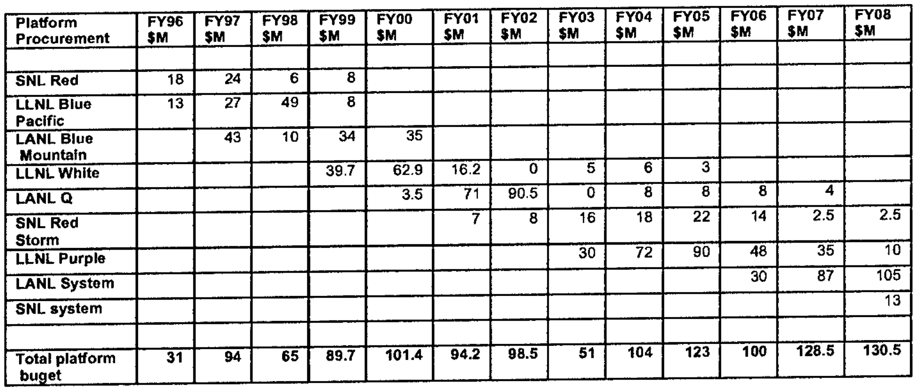
To put these numbers in perspective, we have also calculated an "over-subscription factor" in the last row of the table which is arrived at by taking the ratio of requirements to capacity. Experience in running large user facilities provides a rule of thumb that an over-subscription of over a factor of two is generally indicative of a situation in which there is substantial management risk. From Table 6.2 it can be seen that the over-subscription factor increases to 3.5 in FY06 despite the arrival of the 100TF Purple platform. The original NNSA procurement calls for delivery of a 200TF and 150TF platform in FY07 and FY08 respectively and this brings the over-subscription factor for these years to a more tolerable value of 1.8 and 1.9 respectively. However, it should be noted that the projections for the years FY09 and FY10 (not shown) indicate that the over-subscription factor will rise to 4.7 in FY09 and 6.1 in FY10 in the absence of further computational capability/capacity. We have not included these years in our table as we were not briefed on proposed platform budgets in these years.

6.2 Assessing the Risk of Deferred Acquisition

In order to assess the risk of deferring platform acquisition, we took the original ASCI platform procurement and assumed a $34M delay in funding in FY04. This amount roughly represents 25% of the platform acquisition budget for FY04. What follows is a scenario which serves to illustrate the possible response to such a realignment of funding. It should be stressed that there are certainly other possible ways to respond but we feel that the discussion below is illustrative of the risk factors that emerge.

We have assumed in what follows that the $34M delay is distributed as a $25M delay in funding for the Purple procurement and an $8M delay in funding for the Red Storm procurement. We further assume that such a delay will affect directly the deployment of the Purple C platform (as opposed to BlueGene/L). In this case, it is likely that the deployment of both Red Storm and Purple C could be delayed to FY05 and FY06 respectively. We further assumed the return of funding for Red Storm in FY06 and for Purple C in FY07. A concomitant effect of the return of the funding for these platforms is to increase significantly the procurement budget in these years. A natural assumption is that this will lead to leveling of the budget in future years. Again there are several ways to accomplish this, but, in order to avoid large excursions of the budget in the out years, it will be necessary to delay both the procurement of the 200TF LANL platform and the SNL 150TF platform. A scenario which illustrates all of the above considerations is displayed in Table 6.3.

Table 6.3: Budget With Funding Delay



Platform Procurement	FY96 $M	FY97 $M	FY98 $M	FY99 $M	FY00 $M	FY01 $M	FY02 $M	FY03 $M	FY04 $M	FY05 $M	FY06 $M	FY07 $M	FY08 $M
SNL Red	18	24	6	8									
LLNL Blue Pacific	13	27	49	8									
LANL Blue Mountain		43	10	34	35								
LLNL White				39.7	62.9	16.2	0	5	6	3			
LANL Q					3.5	71	90.5	0	8	8	8	4	
SNL Red Storm						7	8	16	18	22	14	2.5	2.5
LLNL Purple								30	72	90	48	35	10
LANL System											30	87	105
SNL system													13
Total platform buget	31	94	65	89.7	101.4	94.2	98.5	51	104	123	100	128.5	130.5

The associated risk of the particular scenario described above is again best considered in light of its effect on computational capabilities and the ability to meet program requirements. This is shown in Table 6.4.

As a result of the delay in deployment in FY04 and the resulting cascading delay in future years, the over-subscription factor rises significantly in FY04–FY07. Recall that in this metric an over-subscription fraction above two is indicative of increased managerial risk. The implication of this scenario is that required calculations to support upcoming stockpile stewardship milestones may be delayed due to insufficient resources. We therefore conclude that there is indeed increased risk to stockpile deliverables under these assumptions.

Table 6.4: Capability and Capacity With Funding Delay

Platform TF	FY96	FY97	FY98	FY99	FY00	FY01	FY02	FY03	FY04	FY05	FY06	FY07	FY08
SNL Red	0	0	0	3.15	3.15	3.15	3.15	3.15					
LLNL Blue Pacific	0	0	0	3.89	3.89	3.89	3.89	3.89					
LANL Blue Mountain		0	0	3.07	3.07	3.07	3.07	3.07					
LLNL White				0	12.3	12.3	12.3	12.3	12.3	0			
LANL Q					0	0	0	20	20	20	20	20	0
SNL Red Storm						0	0	0	0	40	40	40	40
LLNL Purple								0	26	26	100	100	100
LANL System										0	0	0	200
SNL system											0	0	0
Total platform TF	0	0	0	10.11	22.41	22.41	22.41	42.41	58.3	86	160	160	340
Program Requirements									106	343	565	631	923
Fraction oversubscribed									**1.8**	**4.0**	**3.5**	**3.9**	**2.7**

We summarize the discussion above through a comparative plot shown in Figure 6-1. In the figure, we compare the projected capability as a function of time for the original proposed NNSA procurement and the delayed procurement scenario discussed above. For reference the program requirements are shown in blue, the original plan in pink and the delayed procurement in purple. The "risk threshold" corresponding to an over-subscription factor above two is shown in black. In this case, high risk corresponds to being *below* this curve. It can be seen that, on balance, the effect of delaying computational capability is to increase risk.

The orange line in Figure 6-1 corresponds to an alternative procurement scenario (discussed below) which could in principle significantly mitigate risk by balancing the acquisition of capability platforms with commercial off-the-shelf capacity computing. Prior to discussing this scenario we examine below some aspects of the computational capacity that can be provided with present day cluster technology.

6.3 Capability and Capacity using Commodity Clusters

The advent of commodity architecture (as evidenced by processors such as Intel's Itanium and AMD's Opteron) has made it possible today to achieve computational performance at the level of roughly 1 TF by aggregating 1000 processors and connecting them via a high speed, low latency network such as those provided today, for example by Quadrics or Myricom.

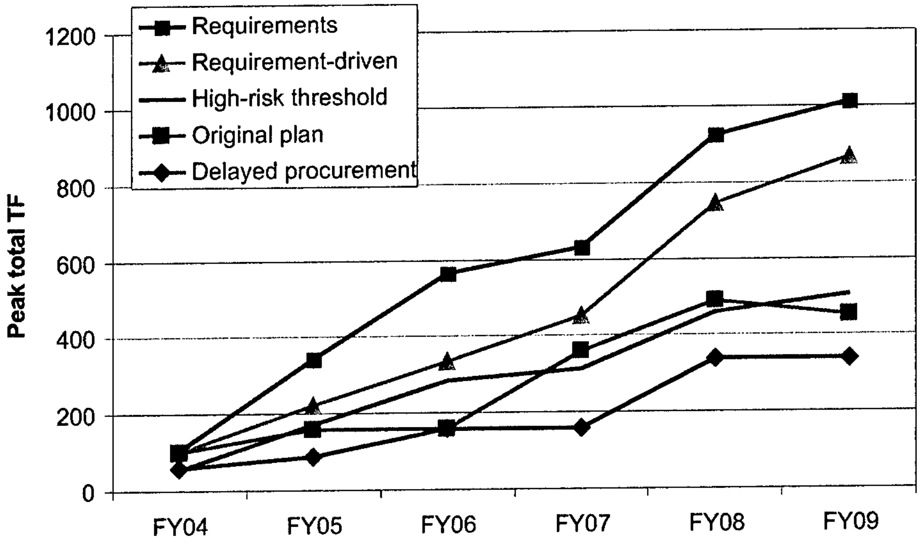


Figure 6-1: Capability and Requirements

This impressive growth in performance is fueled by what is commonly characterized as "Moore's law" which is the observation that the number of transistors on a commodity processor doubles roughly every 1.5 years. This does not mean that actual performance in terms of measured flops increases at this rate. However, it has been observed (cf. Seager presentation) that as a result of Moore's law, performance does appear to increase at the more conservative rate of a factor of 2 every two years. For the purposes of estimation of future performance we will adopt this estimate although it should be noted that continued growth at this rate is in question as discussed in Section 5.5.2. A more conservative growth rate of a factor of 2 every 3 years may in fact reflect better future trends in commodity processor capabilities.

The cost of such clusters is typically in the range of $1M/TF although estimates vary from a low of $0.5M/TF (as presented by Sandia Labs) to a high of $1.5M/TF (as presented by LLNL). Using these numbers we can project the potential capacity and capability available from commodity clusters. We assume an annual capacity budget of $50M. The entries in Table 6.5 below are in TF and represent an aggregate of all the purchased computing power. We have not indicated how this capability might be deployed but anticipate a mix of clusters of roughly 500 to 2000 processors each.

Table 6.5 represents the achievable capacity in TF that can be purchased using commodity clusters assuming that a 1 TF cluster can be purchased for $1M and assuming that the purchased capacity is retired on a five year basis. Similar scenarios

Table 6.5: Capacity Achievable With Commodity Clusters

	2003	2004	2005	2006	2007	2008	2009	2010	2011
$M per TF	1.000	0.707	0.500	0.354	0.250	0.177	0.125	0.088	0.063
TF purchased	0.0	70.7	100.0	141.4	200.0	282.8	400.0	565.7	800.0
Aggregate capacity	0.0	70.7	170.7	312.1	512.1	795.0	1124.3	1589.9	2248.5

can be constructed based on a three year retirement cycle or using a more conservative growth rate for processor performance. As can be seen, if one argues only in terms of aggregate TF, it is possible to meet the program requirements for capacity. Indeed one could argue that the entire program requirements could be met in this way but these estimates are based totally in terms of capacity and would not meet capability needs, which, as we have discussed elsewhere, can credibly exceed a PF. In contrast, the maximum capability of a cluster consisting of 5000 nodes would be in the range of 5–10 TF in FY03 and could be expected to increase to 75–150 TF in FY11 using an optimistic growth rate in speed of 40% per year, or 40-80 TF using the more realistic estimate of 30% per year. In either case this would not meet the requisite capability needs of the program.

Many other such scenarios can be considered but a cursory examination of these results leads to the conclusion:

The programmatic shortfall in terms of capacity can largely be met via an acquisition strategy that utilizes commodity cluster technology.

6.4 A Requirements-Driven Acquisition Scenario

The considerations above imply that cluster technology could be used to meet programmatic capacity requirements. But without further developments in processor and networking technology, it is unlikely that the SSP program's aggressive capability requirements (projected to be over a PF in 2010) would be met. In this section we describe a possible scenario which potentially could address these concerns while providing adequate capacity.

The major components of such a scenario are as follows:

1. The FY04 Purple and Red Storm procurements proceed on schedule in FY04

and a small delay is incurred in BlueGene/L in FY05. This provides respectively 40 and 100 TF of capability as well as capacity in order to meet short term programmatic requirements

2. Beginning in FY05, capacity is purchased for the tri-lab complex using the cluster strategy discussed above. The amount of capacity delivered to each laboratory should be sized according to the workload associated with programmatic deliverables.

3. A "capability exploration"program to accelerate both processor and network development is also initiated in FY05. The ASCI program would invest approximately \$20–30M per year to explore promising new processor and networking architectures such as those discussed in Section 5.5 with the goal of achieving a PF in 2010. Such a program would have as its immediate goal the delivery of an intermediate 200TF capability platform by FY08 to assess scalability of ASC codes in preparation for delivery of a full PF capability modeling platform in 2010. Such an acquisiton must be carefully guided by performance of ASCI applications.

4. The "capability exploration"investment continues after delivery of the PF capability so as to facilitate future advances leading to multi-PF capability beyond 2010.

We present a spending profile as well as the projected delivered TF in Tables 6.6 and 6.7 below. As can be seen, such a scenario leads to a spending profile that is similar to the funding level proposed in the original NNSA procurement. The deployment of significant capacity via commodity clusters leads to a healthy balance of capacity and capability while staying well above the high risk threshold curve (as shown in Figure 6-1).

Note that in this approach both capability and capacity are centrally managed in that identical (and interchangeable) cluster capacity computing is delivered to all three laboratories. The advantage of this is that the combined throughput then becomes uniformly available to the entire tri-lab complex allowing for flexible demand-driven management of resources.

Table 6.6: ASCI Platform Procurement Budget With Commodity Capacity

	FY96 $M	FY97 $M	FY98 $M	FY99 $M	FY00 $M	FY01 $M	FY02 $M	FY03 $M	FY04 $M	FY05 $M	FY06 $M	FY07 $M	FY08 $M	FY09 $M
SNL Red	18	24	6	8										
LLNL Blue Pacific	13	27	49	8										
LANL Blue Mountain		43	10	34	35									
LLNL White				39.7	62.9	16.2	0	5	6	3				
LANL Q					3.5	71	90.5	0	8	8	8	4		
SNL Red Storm						7	8	16	26	22	6	2.5	2.5	
LLNL Purple								30	97	70	68	10	10	
Tri lab capacity									0	30	40	30	20	20
Path forward arch phase 1										10	20			
Tri lab capability												30	60	80
Path forward arch phase 2												30	30	30
Total platform budget	**31**	**94**	**65**	**89.7**	**101.4**	**94.2**	**98.5**	**51**	**137**	**143**	**142**	**106.5**	**122.5**	**130**

Table 6.7: Capability and Capacity With Commodity Capacity

Platform TF	FY96	FY97	FY98	FY99	FY00	FY01	FY02	FY03	FY04	FY05	FY06	FY07	FY08	FY09
SNL Red	0.0	0.0	0.0	3.2	3.2	3.2	3.2	3.2						
LLNL Blue Pacific	0.0	0.0	0.0	3.9	3.9	3.9	3.9	3.9						
LANL Blue Mountain		0.0	0.0	3.1	3.1	3.1	3.1	3.1						
LLNL White				0.0	12.3	12.3	12.3	12.3	12.3	0.0				
LANL Q					0.0	0.0	0.0	20.0	20.0	20.0	20.0	20.0	0.0	
SNL Red Storm						0.0	0.0	0.0	40.0	40.0	40.0	40.0	40.0	
LLNL Purple								0.0	26.0	100.0	100.0	100.0	100.0	100.0
Tri lab capacity									0.0	60.0	173.1	293.1	406.3	566.3
Tri lab capability											0.0	0.0	200.0	200.0
Total platform TF	0.0	0.0	0.0	10.1	22.4	22.4	22.4	42.4	98.3	220.0	333.1	453.1	746.3	866.3
Program Requirements									106.0	343.0	565.0	631.0	923.0	1010.0
Fraction oversubscribed									**1.1**	**1.6**	**1.7**	**1.4**	**1.2**	**1.2**

7 CONCLUSIONS AND RECOMMENDATIONS

This section summarizes the conclusions and recommendations of the 2003 JASON summer study commissioned by the National Nuclear Security Administration (NNSA) to identify the distinct requirements of its stockpile stewardship program (SSP) in relation to the hardware procurement strategy of the Advanced Simulation and Computing (ASCI) program.

Today, ASCI comprises high-performance computing hardware, suites of large codes built on a validated scientific/engineering base, experienced people and connections to the greater scientific, computing and national security communities. ASCI is now one of the pillars of the SSP, and the tools and methods developed under ASCI have evolved to the point where they are today *essential* to stockpile stewardship. Large reductions in the computing budget of ASCI would pose substantial risks, and could lead to more costly alternatives when it comes time to certify the stockpile. The people supported through ASCI are a substantial asset, who work closely with NW designers, engineers and managers, and have acquired invaluable expertise in developing and optimizing ASCI tools and in establishing and improving the scientific credibility of NW simulations.

ASCI contributes substantially to stockpile stewardship by enabling qualitatively new capabilities for modeling weapons physics and by providing guidance to scientists and managers in support of SSP milestones. The scientific modeling taking place within ASCI contributes to confidence in the stockpile as well as to continuing improvement in the physics base of the models and the codes which express them. We have determined that increased computational capability is needed for a sustainable and credible program over the long term. "Spin-off" benefits of ASCI, such as training a cadre of US scientists and computer scientists in real-world problems of high-performance computing and attracting excellent young people to work at the national laboratories, are being realized. With the insight developed over the past few years it is now appropriate to evaluate future computing technologies for their relevance to the United States in maintaining a safe and reliable nuclear deterrent based on simulations of its stockpile.

Determining a quantitative measure for the performance requirements of ASCI computing is difficult, as there is no single metric that can precisely and completely describe the performance requirements for computers of the class needed for ASCI. Users—designers, scientists and engineers—place great value on time-to-solution, the interval of "wall-clock" time it takes to prepare and execute a given computation. Large systems, such as ASCI platforms, are advertised as having 1 or 10 or 100 peak-

TeraFLOPS (TF), a figure that represents the combined floating-point processing speed of all CPUs comprising the system. Except for the most trivial codes (and none of the ASCI codes are trivial), it is impossible to deliver all of the aggregate processing power to a given computation. The amount of processing power that can be delivered to a given computation is called its *efficiency* and is a function of CPU performance, as well as memory-access latency and bandwidth, interconnection network performance and other constraints. ASCI applications are particularly demanding in this respect, both due to their size and to the requirements of irregular grids. It is not yet completely known what processing speed might ultimately be sufficient for fully and accurately simulating the complex physics processes in a nuclear weapon. Future predictive simulation will most likely involve a combination of "subgrid models" based on weapons science phenomenology and experiment, along with directly simulated phenomena. Such simulations will still require significant computational resources both to resolve the physics and to capture the geometric complexity of modern weapons systems. It is also clear that improved physics understanding and better algorithms will play a key role in ASCI meeting its goals, since it is impractical to simulate an entire weapon "from button to bang" from first principles.

7.1 Computing Requirements

Two commonly used measures of the overall productivity of ASCI platforms are *Capability* and *Capacity*. Despite the inadequacies indicated above (and described more fully in the body of this report), the common unit of measure for both is peak floating-point operations, noted as TeraFLOPS or TF (10^{12} operations per second). Used in this context, *Capability* refers to the maximum processing power possible that can be applied to a single job and *Capacity* represents the total processing power available from all machines capable of operating ASCI codes. A given amount of *Capability* implies *Capacity* in two ways: 1) by its direct contribution to total capacity and 2) because a high-*Capability* machine can be re-configured into multiple lower-*Capability* machines to run multiple shorter jobs, often with somewhat improved overall performance.

The distinct requirements of the ASCI program for *Capability* and *Capacity* flow from the technical objectives of SSP. The SSP work is organized under eight main task areas:

Directed Stockpile Work (DSW). Evaluation, maintenance and refurbishment of the stockpile supporting the annual certification process. Includes life-extension programs (LEPs) to manage aging issues in specific weapons systems. Requires a broad range of both computing capability and capacity.

Campaigns. Science and engineering efforts cutting across weapons types aimed at improving understanding of specific weapons physics and engineering issues. Demands extensive computing capability and capacity for simulations.

Significant Finding Investigations (SFIs). Technical investigations of anomalies uncovered by surveillance of disassembled stockpile weapons as well as non-destructive testing of complete weapons. Demands a broad range of computing capabilities and capacity for simulations.

Baselining. Systematic and consistent modeling using ASCI codes of the full set of underground test data for a given weapon system. Requires range of calculations, the largest of which involve significant computational capability at the limit of knowledge of weapons science.

Safety. Engineering calculations over a broad range of parameters describing potential accident scenarios.

Stockpile-to-Target Sequence (STS) Requirements. Modeling and simulation of all possible environments encountered during the delivery of weapons. Demands a broad range of computing capability and capacity for simulations.

Support to production. ASCI simulations related to manufacturing, assembly and disassembly of weapons. Typically these involve modest computing demands but with some requirements for significant capability.

Surety. Use-control and other classified aspects of the stockpile. Moderate to large demands on ASCI capabilities, but can present a significant demand on computing capacity.

Today, the ASCI platforms of highest *Capability* are LLNL's "White" at 12.3 TF and LANL's "Q" at 20 TF. The next planned acquisitions are SNL's "Red Storm" projected to be 40 TF and LLNL's "Purple C" at 100 TF. SSP requirements over the next few years call for a few large jobs which need the largest available *Capability* but the most important trend is a factor-of-two oversubscription in ASCI *Capacity* which is projected to go on and potentially worsen in the foreseeable future. This level of demand means that jobs are selected to run by some combination of administrative fiat and management priority-setting; it creates strong incentives for all involved in ASCI computing to improve performance of algorithms and platforms for greater delivered performance. A strong and valuable effort has been made by the ASCI program to increase the efficiency of performance (ratio of computing operations delivered to peak performance) to current levels; in practice, about 0.5–15% of the peak processing speed is realized, depending on algorithms and details of implementation. Similar efficiencies are found in many commercial, engineering and scientific applications. While applauding efforts within ASCI to improve efficiency, a continuing investment in improving efficiency is called for; these improvements will come from both improved hardware but also from improved algorithms and system software (such as better compilers, the use of automatic code generators, and improved inter-processor communication mechanisms).

JASON understands that there are difficult issues involved with allocating access to the most capable machine. Even so, it is important that this machine be used for large computations that require *Capability* and not simply divided up into a large *Capacity* machine. It must be available for large end-to-end burn simulations, while many of the day-to-day computations could be satisfied through *Capacity* computing. It is also very important that the computer scientists be given the opportunity to evaluate their improved algorithms in terms of scalability as the machine. In light of the allocation issues, we recommend that a committee be established to mete out access to the *Capability* machine. This is similar to how other large scientific instruments such as accelerators are managed. In order to avoid paralysis, we recommend that a Director be appointed, who is empowered to make the decisions on advice from the committee.

Within 10 years, estimates of the demand for *Capability* and general physics arguments indicate a machine of 1000 TF = 1 PetaFLOP (PF) will be needed to execute the most demanding jobs. Such demand is inevitable; it should not be viewed, however, as some plateau in required *Capability*—there are sound technical reasons to expect even greater *Capability* demand in the future.

The recent anointing (in the "Top 500" listings) of the Earth Simulator as the world's most powerful computer has raised concerns about the choice of hardware by the ASCI program. We have investigated the relevance of this ranking to ASCI problems. On the basis of performance estimates for unclassified stand-ins for ASCI codes (LA UR03-1263) we conclude that the Earth Simulator's performance (if used as an ASCI machine) would lie between those of the nominal 20 TF ASCI Q machine and the originally planned 30 TF ASCI Q machine. The Earth Simulator's cost has been reported to be $350,000,000, more than twice that of ASCI Q and nearly three times that of ASCI White.

7.2 Recommendations

In evaluating NNSA's current ASCI platform acquisition strategy, we see two areas of substantial risk. The first is related to the current and projected oversubscription in *Capacity*. A factor-of-two oversubscription is probably manageable; however, larger demand on *Capacity* could become unmanageable. The likely result would be the delay in meeting SSP technical milestones and/or overly-cautious decision-making leading to expensive—at the $50M–$100M level—mitigation programs from elsewhere in the SSP (for addressing LEP or SFI issues, for example) that might not be necessary if sufficient confidence were provided by timely simulations. The second area of risk is the lack of a credible "road map" to acquiring the next generation of machines with PF *Capability*, which will be needed within a decade. Scaling to 1 PF using present machine architectures implies very large numbers of processors—of order 100,000, perhaps—might be needed. Such large numbers raise serious questions of scalability of code performance and of machine reliability.

JASON recommends SSP management consider four general areas for mitigating the risks associated with its present ASCI acquisition strategy:

1. Platform Acquisition:

   (a) Modify the allocation of resources in the current acquisition plan to provide additional *Capacity* platforms starting as soon as possible.

   (b) Lay the groundwork for future *Capability* machines. This may involve acquisition of "*Capability*-exploration" machines focused on optimizing efficiency in computation for ASCI problems in order to gain experience with architectures that might plausibly be extended to the PF level. These acquisitions can probably begin in the FY 06–07 time frame and may be able to replace currently planned greater-than 100 TF-scale machines that are scheduled to follow the Purple C and Red Storm acquisitions.

2. SSP Requirements:

   Set priorities in the SSP requirements and assign ASCI resources accordingly. This is essential to reduce excess computing demand and to assure that the high priority problems are addressed to meet goals as scheduled. In particular, some of the STS requirements identified for Cold War scenarios place significant demands on ASCI resources (and, in fact, on the entire SSP). They should be reviewed carefully in the context of current and anticipated US security needs. Those judged to no longer meet a compelling cost/benefit standard should be either relaxed or assigned an appropriately lower ranking in the queue of high-priority tasks for ASCI's available and planned future resources.

3. ASCI Operations:

   (a) Expand access to ASCI "most-capable" systems to best align ASCI *Capability* with overall SSP priorities and needs. We applaud efforts to make ASCI platforms available to workers across the complex, regardless of where a given machine or work-group is located, but present quotas should be re-examined. This might involve adopting priority-setting methods similar to those used elsewhere in the scientific community where key resources are over-subscribed. One example would be an ASCI allocation "czar", assisted by a program advisory committee (PAC). We expect additional benefits to accrue from enhanced scientific communications and competition. Depending on the architecture of future PF-level machines, it is possible that future acquisitions will entail fewer high-*Capability* platforms, in which case convenient and flexible access across the complex with crisp, transparent manangement of all computing resources becomes essential.

   (b) Continue and expand, as appropriate, investment in computational science investigations directed toward improving the delivered performance of algorithms running on ASCI platforms; consider dedicating regularly scheduled machine time on capability platforms for efficiency studies.

4. Nuclear Weapons Science:

   Encourage the advance of NW science at every opportunity in the SSP and ASCI programs. Better science is the most cost-effective way to reduce risk in the stewardship program and the only possible way to achieve sufficiency in the modeling and understanding of these complex systems. Some excellent new science is beginning to emerge in association with ASCI. In the body of this report, we comment on some of this new science and suggest possible extensions.

We have determined that there is a significant risk to SSP and to the scientific program if the proposed cuts are made to the ASCI aquisitions budget. To assess this additional risk, we constructed an acquisition scenario where FY 04 funding was reduced by $33M (24%), requiring that the procurements of Red Storm and Purple C be stretched out. Given the approximately $50M shortfall in requested FY 03 funds, the FY 04 reduction was assumed to lead to further reductions in FY 05, 06 and 07, which we modeled by an approximately flat acquisition budget through FY 08 at a level of approximately $120M–$130M. The net effect of such a stretch-out is to reduce overall *Capacity* to below 1/3 of demand during the critical program years FY 05–08. In this period, several program milestones are to be completed including the refurbishment and first production units of one major weapons system, and the qualification of critical components of another. We judge the risk to SSP of such a delay to be *high*, not so much from the delayed *Capability*, but from the very serious reduction in overall *Capacity*. The large jobs projected to require 100 TF-level *Capability* might be deferred at moderate risk to the program, but the resulting very large oversubscription in *Capacity* risks becoming unmanageable. Purchasing the proposed large platforms—Purple C and Red Storm—on a stretched-out, suboptimal schedule where their CPUs and other components are bought after their performance/cost prime strikes us as being unwise both in terms of delivered capability and capacity.

To summarize, our major concern is in providing as soon as practical much needed capacity to the ASCI program lest the resulting very large oversubscription in *Capacity* becomes unmanageable. In addition, a road map must be developed to deliver to the program machines of the requisite *Capability*.

A APPENDIX: BRIEFERS

Dr. David Crandall	DOE Defense Programs
Dr. William H. Reed	U. S. Department of Energy
Mr. Charles F. McMillan	Lawrence Livermore National Labs
Mr. Rodney B. Wood-Schultz	Los Alamos National Laboratory
Mr. James S. Peery	Los Alamos National Laboratory
Mr. David E. Womble	Sandia National Laboratories, New Mexico
CAPT Douglas Coppinger	U.S. Strategic Command
Mr. Stephen J. Rottler	Sandia National Laboratories, New Mexico
Mr. Stephen E. Lott	Sandia National Laboratories, New Mexico
Mr. Mark K. Seager	Lawrence Livermore National Laboratory
Mr. Peter W. Rambo	Lawrence Livermore National Laboratory
Ms. Rose M. Baltrusaitis	Los Alamos National Laboratory
Ms. Juliana J. Hsu	Lawrence Livermore National Laboratory
Ms. Elaine C. Flower-Maudlin	Los Alamos National Laboratory
Mr. Thomas C. Bickel	Sandia National Laboratories, New Mexico
Mr. Thomas L. McAbee	Lawrence Livermore National Laboratory
Mr. Laurence E. Fried	Lawrence Livermore National Laboratory
Mr. Francis L. Addessio	Los Alamos National Laboratory
Mr. Mark B. Chadwick	Los Alamos National Laboratory
Mr. Larry G. Wiley	Lawrence Livermore National Laboratory
Mr. Omar A. Hurricane	Lawrence Livermore National Laboratory
Mr. John B. Aidun	Sandia National Laboratories, New Mexico
Mr. Mark A. Hedeman	Sandia National Laboratories, New Mexico
Ms. Dana A. Knoll	Los Alamos National Laboratory
Mr. Brian S. Pudliner	Lawrence Livermore National Laboratory
Mr. Jim E. Morel	Los Alamos National Laboratory
Mr. Michael R. Zika	Lawrence Livermore National Laboratory
Mr. Robert P. Weaver	Los Alamos National Laboratory
Mr. Leonard Lorence	Sandia National Laboratories, New Mexico
Dr. Joseph C. Martz	Los Alamos National Laboratories
Mr. Bill Moran	Lawrence Livermore National Laboratory
Mr. Martin Pilch	Sandia National Laboratories, New Mexico
Mr. Harvey J. Wasserman	Los Alamos National Laboratory
Mr. Robert W. Leland	Sandia National Laboratories, New Mexico
Mr. Charles W. Nakhleh	Los Alamos National Laboratories
Mr. Langdon S. Bennett	Los Alamos National Laboratories
Mr. John M. Scott	Los Alamos National Laboratory
Mr. Joel S. Lash	Sandia National Laboratories, New Mexico
Mr. Dean Dobranich	Sandia National Laboratories, New Mexico
Mr. Charles P. Verdon	Lawrence Livermore National Laboratory
Mr. Burton Smith	Cray, Inc.
Mr. Thomas O. Hunter	Sandia National Laboratories
Dr. Ray Juzaitis	Los Alamos National Laboratory
Mr. Stu Feldman	IBM
Mr. Bruce Goodwin	Lawrence Livermore National Laboratory
Dr. Francis Sullivan	IDA/CCS

DISTRIBUTION LIST

Director of Space and SDI Programs
SAF/AQSC
1060 Air Force Pentagon
Washington, DC 20330-1060

CMDR & Program Executive Officer
U S Army/CSSD-ZA
Strategic Defense Command
PO Box 15280
Arlington, VA 22215-0150

DARPA Library
3701 North Fairfax Drive
Arlington, VA 22203-1714

Assistant Secretary of the Navy
(Research, Development & Acquisition)
1000 Navy Pentagon
Washington, DC 20350-1000

Principal Deputy for Military Application [10]
Defense Programs, DP-12
National Nuclear Security Administration
U.S. Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

Superintendent
Code 1424
Attn: Documents Librarian
Naval Postgraduate School
Monterey, CA 93943

DTIC [2]
8725 John Jay Kingman Road
Suite 0944
Fort Belvoir, VA 22060-6218

Strategic Systems Program
Nebraska Avenue Complex
287 Somers Court
Suite 10041
Washington, DC 20393-5446

Headquarters Air Force XON
4A870 1480 Air Force Pentagon
Washington, DC 20330-1480

Defense Threat Reduction Agency
Attn: Dr. Arthur T. Hopkins [12]
6801 Telegraph Road
Alexandria, VA 22310

IC JASON Program [2]
Chief Technical Officer, IC/ITIC
2P0104 NHB
Central Intelligence Agency
Washington, DC 20505-0001

JASON Library [5]
The MITRE Corporation
WA549
7515 Colshire Drive
McLean, VA 22102

U. S. Department of Energy
Chicago Operations Office Acquisition and
Assistance Group
9800 South Cass Avenue
Argonne, IL 60439

Dr. Allen Adler
Director
DARPA/TTO
3701 N. Fairfax Drive
Arlington, VA 22203-1714

Dr. Jane Alexander
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Dr. A. Michael Andrews
Director of Technology
SARD-TT
Room 3E480
Research Development Acquisition
103 Army Pentagon
Washington, DC 20310-0103

Dr. William O. Berry
Director
Basic Research ODUSD(ST/BR)
4015 Wilson Blvd
Suite 209
Arlington, VA 22203

Mr. David Bowman
Los Alamos National Laboratory
Mailstop D411
Los Alamos, NM 87545

Dr. Albert Brandenstein
Chief Scientist
Office of Nat'l Drug Control Policy Executive
Office of the President
Washington, DC 20500

Ambassador Linton F. Brooks
Under Secretary for Nuclear Security/
Administrator for Nuclear Security
1000 Independence Avenue, SW
NA-1, Room 7A-049
Washington, DC 20585

Dr. Steve Buchsbaum
DARPA/STO
3701 N. Fairfax Drive
Arlington, VA 22203-1714

Dr. Darrell W. Collier
Chief Scientist
U. S. Army Space & Missile Defense Command
PO Box 15280
Arlington, VA 22215-0280

Dr. James F. Decker
Principal Deputy Director
Office of the Director, SC-1
Room 7B-084
U.S. Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

Dr. Patricia M. Dehmer [5]
Associate Director of Science for Basic Energy
Sciences, SC-10
Office of Science
U.S. Department of Energy
19901 Germantown Road
Germantown, MD 20874

Ms. Shirley Derflinger [15]
Technical Program Specialist
Office of Biological & Environmental
Research, SC-70
Office of Science
U.S. Department of Energy
19901Germantown Road
Germantown, MD 20874

Dr. Martin C. Faga
President and Chief Exec Officer
The MITRE Corporation
Mail Stop N640
7515 Colshire Drive
McLean, VA 22102

Mr. Dan Flynn [5]
Program Manager
DI/OTI/SAG
5S49 OHB
Washington, DC 20505

Ms. Nancy Forbes
Senior Analyst
DI/OTI/SAG 5S49 OHB
Washington, DC 20505

Dr. Paris Genalis
Deputy Director
OUSD(A&T)/S&TS/NW
The Pentagon, Room 3D1048
Washington, DC 20301

Mr. Bradley E. Gernand
Institute for Defense Analyses
Technical Information Services
Room 8701
4850 Mark Center Drive
Alexandria, VA 22311-1882

Dr. Lawrence K. Gershwin
NIC/NIO/S&T
2E42, OHB
Washington, DC 20505

Brigadier General Ronald Haeckel
U.S. Dept of Energy
National Nuclear Security Administration
1000 Independence Avenue, SW
NA-10 FORS Bldg
Washington, DC 20585

Dr. Theodore Hardebeck
STRATCOM/J5B
Offutt AFB, NE 68113

Dr. Robert G. Henderson
Director, JASON Program Office
The MITRE Corporation
7515 Colshire Drive
Mail Stop W950
McLean, VA 22102

Dr. Bobby R. Junker
Office of Naval Research
Code 31
800 North Quincy Street
Arlington, VA 22217-5660

Dr. Andrew F. Kirby
DO/IOC/FO
6Q32 NHB
Central Intelligence Agency
Washington, DC 20505-0001

Dr. Anne Matsuura
Army Research Office
4015 Wilson Blvd
Tower 3, Suite 216
Arlington, VA 22203-21939

Dr. Maureen I. McCarthy
Homeland Security
Anacostia Naval Annex
Building 410
250 Murray Lane, SW
Washington, DC 20509

Mr. Gordon Middleton
Deputy Director
National Security Space Architect
PO Box 222310
Chantilly, VA 20153-2310

Dr. Thomas Meyer
DARPA/ATO
3701 N. Fairfax Drive
Arlington, VA 22203

Dr. Julian C. Nall
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

Dr. C. Edward Oliver [5]
Associate Director of Science for Advanced
Scientific Computing Research, SC-30
U.S. Department of Energy
19901Germantown Road
Germantown, MD 20874

Mr. Raymond L. Orbach
Director, Office of Science
U.S. Department of Energy
1000 Independence Avenue, SW
Route Symbol: SC-1
Washington, DC 20585

Dr. Ari Patrinos [5]
Associate Director
Biological and Environmental Research
SC-70
US Department of Energy
19901 Germantown Road
Germantown, MD 20874-1290

Dr. John R. Phillips
Chief Scientist, DST/CS
2P0104 NHB
Central Intelligence Agency
Washington, DC 20505-0001

Records Resource
The MITRE Corporation
Mail Stop W115
7515 Colshire Drive
McLean, VA 22102

Dr. John Schuster
Submarine Warfare Division
Submarine, Security & Tech
Head (N775)
2000 Navy Pentagon, Room 4D534
Washington, DC 20350-2000

Dr. Ronald M. Sega
DDR&E
3030 Defense Pentagon,
Room 3E101
Washington, DC 20301-3030

Mr. Anthony J. Tether
DIRO/DARPA
3701 N. Fairfax Drive
Arlington, VA 22203-1714

Dr. George W. Ullrich [3]
OSD [ODUSD(S&T)]/WS
Director for Weapons Systems
3080 Defense Pentagon
Washington, DC 20301-3080

Dr. Bruce J. West
FAPS
Senior Research Scientist
Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

Dr. Linda Zall
Central Intelligence Agency
DS&T/OTS
3Q14, NHB
Washington, DC 20505-00